



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

1973

# A queueing MODEL simulation with dynamic graphical display.

Riddell, Cyrus Michael.

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/16769>

---

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

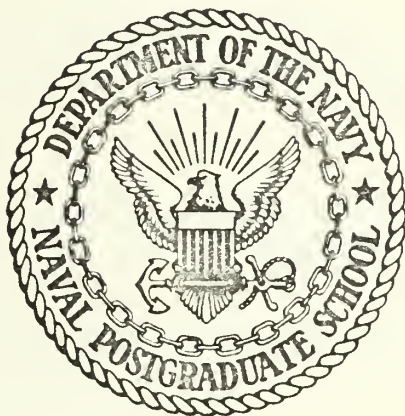
A QUEUING MODEL SIMULATION  
WITH DYNAMIC GRAPHICAL DISPLAY

Cyrus Michael Riddell

Library  
Naval Postgraduate School  
Monterey, California 93940

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

A QUEUING MODEL SIMULATION  
WITH DYNAMIC GRAPHICAL DISPLAY

by

Cyrus Michael Riddell

Thesis Advisor:

R. W. Butterworth

March 1973

*Approved for public release; distribution unlimited.*

T153548



A Queuing Model Simulation  
with Dynamic Graphical Display

by

Cyrus Michael Riddell  
Lieutenant, United States Navy  
B.A., University of Washington, 1965

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL  
March 1973

---



## ABSTRACT

The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.



## TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	7
II.	QUEUEING MODEL SIMULATION - - - - -	9
A.	THE QUEUEING MODELS - - - - -	10
1.	G/G/C- - - - -	10
2.	G/G/C with Losses- - - - -	11
3.	G/G/C with Finite Source - - - - -	12
4.	G/G/C with Feedback to the Queue - - - - -	13
5.	Two Simultaneous G/G/C Queues- - - - -	14
B.	ARRIVAL/SERVICE DISTRIBUTIONS AND QUEUE DISCIPLINES- - - - -	14
1.	Distributions- - - - -	15
a.	K-ERLANG - - - - -	15
b.	Hyperexponential - - - - -	15
c.	Degenerate - - - - -	16
2.	Queueing Disciplines- - - - -	16
a.	FIFO - - - - -	16
b.	LIFO - - - - -	16
c.	SSTF - - - - -	16
d.	Random Draw- - - - -	17
3.	Length of the Simulation - - - - -	17
III.	THE METHOD - - - - -	18
A.	THE COMPUTER SIMULATION- - - - -	18
B.	THE GRAPHICAL DISPLAY- - - - -	18
IV.	STATISTICS FROM THE SIMULATION - - - - -	20



V.	PARAMETER INPUT-	22
A.	SAMPLE INPUT CARD-	22
B.	NAMelist INPUT VIA THE GRAPHICS TERMINAL -	23
C.	INPUT VARIABLES-	23
VI.	THE PROGRAM-	25
	APPENDIX A - ANALOG SCHEMATIC-	28
	COMPUTER PROGRAM LISTING -	29
	INITIAL DISTRIBUTION LIST-	87
	FORM DD 1473 -	89



## LIST OF PICTURES

1.	G/G/C Queuing System - - - - -	10
2.	G/G/C Queuing System with Losses - - - - -	11
3.	G/G/C Queuing System with Finite Source- - - - -	12
4.	G/G/C Queuing System with Feedback - - - - -	13
5.	Two Simultaneous G/G/C Queuing Systems - - - - -	14
6.	Statistics Page- - - - -	20
7.	Parameters Page- - - - -	22



## ACKNOWLEDGEMENTS

The Hybrid Computer at the Naval Postgraduate School is a self-operated system; because many of the complexities of the system were unfamiliar to the programmer, the technical assistance of Mr. R. Limes, Mr. W. Thomas, and Mr. A. Wong was invaluable.



## I. INTRODUCTION

A queuing system is comprised of action facilities (servers) and waiting facilities (queues or lines). The relationship between the components can be of many forms, each with their own characteristics. Consider a tool crib in a machine shop where the machinists are obliged to check out their tools; or consider the telephone system, in which callers request communication capacity. Both situations involve requests for action from limited resources, in these cases, servers or telephone circuits. In order to allocate resources properly, it is necessary to understand the interaction between requests for action and the actions themselves. Individually these can be modeled as stochastic processes, but in order to predict flow through the queuing system, it is necessary to understand the interrelation between the patterns of event occurrences. When demands occur at regular intervals and actions are of fixed duration, it is relatively easy to predict flow; if there are more requests per unit of time than actions over the same time span, the queue will grow without bound unless the event relationship is changed. When variation is added to the timing of events, the flow is not as obvious. For example, if the average number of requests for action equals the average number of actions, per unit of time, but both occur with variability, then it would be useful to be able to predict such quantities as expected delay and action



facility idle time. Queuing models are used for this purpose.

The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.

This is done through a combination of a computer simulation and a computer-driven graphical display.



## II. QUEUEING MODEL SIMULATION

This computer program simulates queueing systems in two stages. First, a next-event type computer simulation creates a time history of the queueing system. Second, a dynamic graphical display depicts the occurrence of the queueing process in real time.

Five basic, academically interesting, queueing models, each with parameter options, can be simulated. Queueing models are generally described in terms of the distribution of request occurrences (arrival distribution), the distribution of action occurrences (service distribution), and the number of action facilities (servers). Specifically, the times between request occurrences are taken to be independent and identically distributed realizations from the arrival distribution. The duration of each action occurrence is also taken to be an independent and identically distributed realization from the service distribution. Variations on these assumptions occur for some models and are described in Section A.

The abbreviation G/G/C describes a queueing system in which the arrival and service distributions are general (unspecified) and there are C servers.

The five options in this simulation are:

- (1) G/G/C
- (2) G/G/C with losses
- (3) G/G/C with feedback



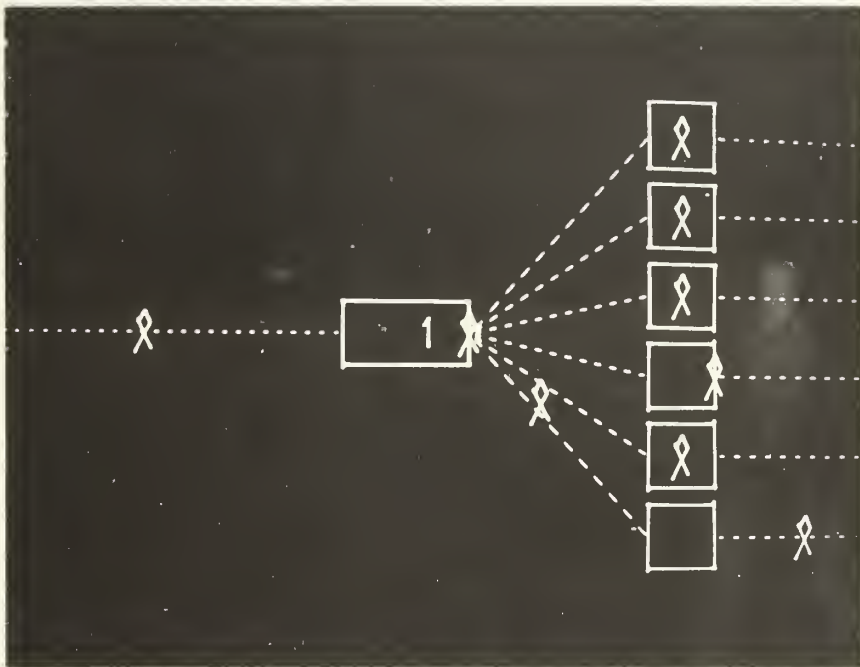
(4) G/G/C with finite source

(5) Two simultaneous G/G/C queues for comparison.

The specific choices of arrival distribution and service distribution which can be simulated are fully described in Section B.

#### A. THE QUEUING MODELS

##### 1. G/G/C



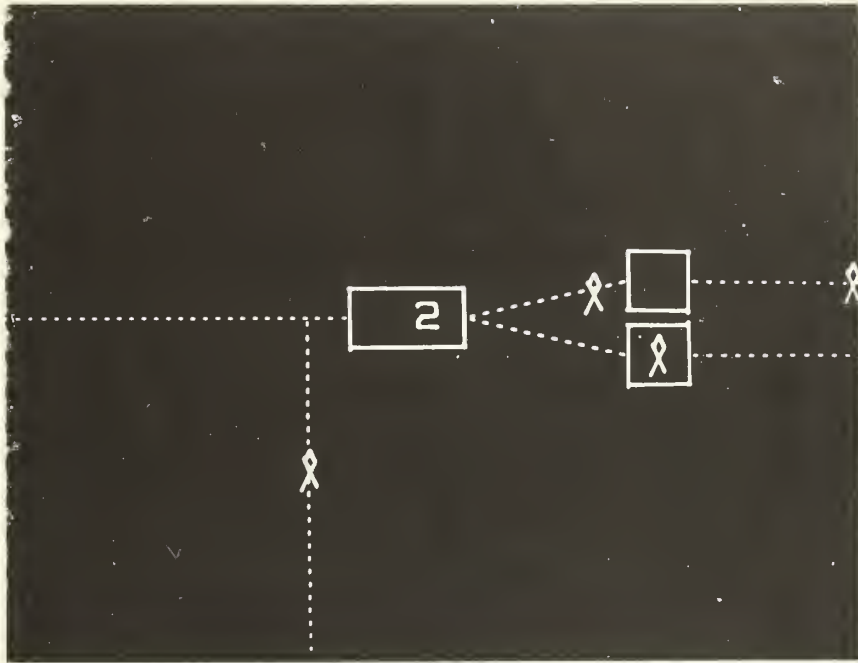
Picture 1

This is the simplest of the queuing models considered here. Each arrival waits in the queue until a server is available, enters service for a calculated period of time, then departs the system.

This model may be simulated with from one to six servers.



## 2. G/G/C with Losses

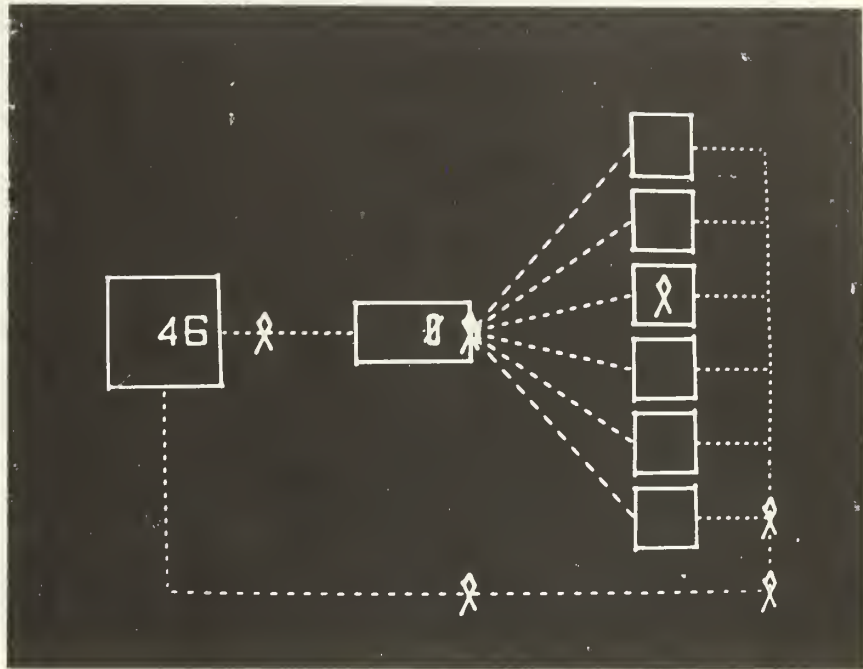


Picture 2

Consider making a long distance telephone call on Mother's Day. If all the trunk lines are busy, your request for service will be rejected. The G/G/C model with losses reflects this by rejecting arrivals who find the queue full. The queue limit used is specified as an input parameter. Note that rejected arrivals are lost and are not assumed to return later. The fraction of all arrivals who are served (not rejected) is recorded for subsequent display.



### 3. G/G/C with Finite Source



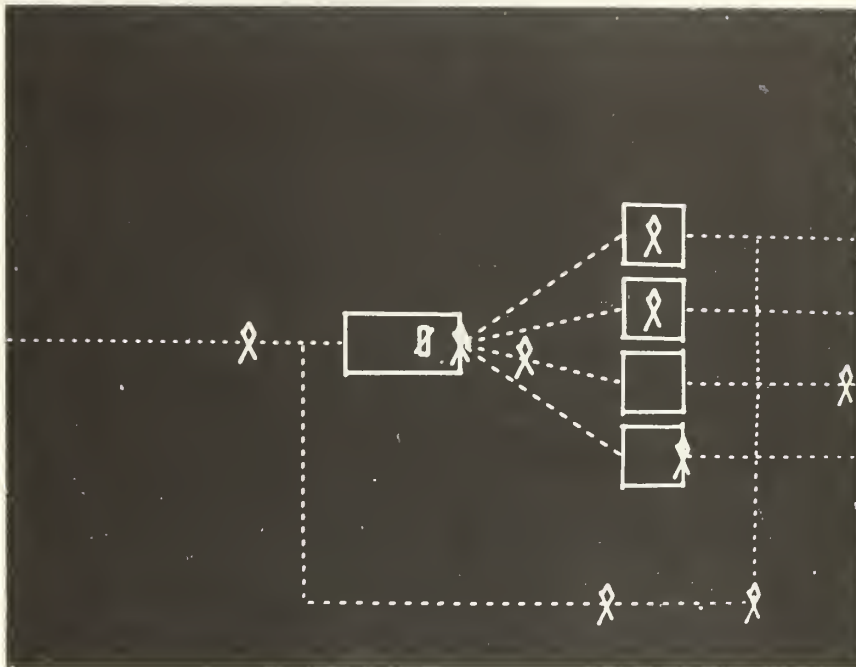
Picture 3

Consider a closed-circuit educational television system at a school. The finite population consists of all the televisions in the system. When one fails, it enters the queue to be serviced. After repair, it re-enters the population pending the next failure.

This situation is modeled by assuming that each action source, upon entering the idle population, is assigned a waiting time from the arrival distribution. When this time expires, the source requests service, joining the queue if necessary. As each action source terminates its service, it returns to the idle population. The user specifies the number of servers (one to six) and the size of the population.



#### 4. G/G/C with Feedback to the Queue

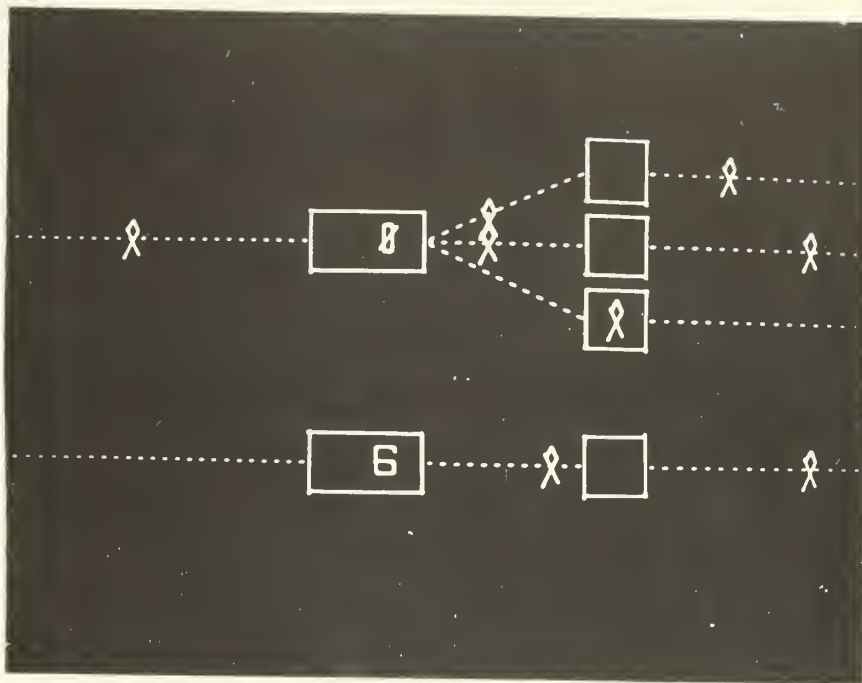


Picture 4

This is an extension of the G/G/C model, with from one to six servers, allowing customers leaving service to depart the system or to return to the queue. At each service completion, the customer leaves the system with probability  $P$  or returns to the queue with probability  $1-P$ . The user specifies  $P$  as an input option.



## 5. Two Simultaneous G/G/C Queues



Picture 5

This option allows comparison of two standard G/G/C queues, each with up to three servers. All parameters of each system are specified separately, and, if desired, identical arrival streams may be utilized. This allows comparisons between different arrival and/or service distributions and associated parameters, and between a different number of servers. Also, the effect of the various queuing disciplines can be compared.

### B. ARRIVAL/SERVICE DISTRIBUTIONS AND QUEUE DISCIPLINES

As well as options already described, the following arrival and service distributions are available to the user of the program. Any of the following probability



distributions can be used as an arrival or service distribution.

1. Distributions

a. K - ERLANG (exponential when K=1)

$$F(t) = 1 - \sum_{J=0}^{K-1} \frac{(\beta t)^J}{J!} e^{-\beta t}$$

$$\text{RATE } \lambda = \frac{\beta}{K}$$

$$CV = \frac{1}{\sqrt{K}}$$

Input parameters are:

(1) K

(2) Rate - in terms of mean number of events per minute.

b. Hyperexponential

$$F(t) = p(1 - e^{-\lambda_1 t}) + q(1 - e^{-\lambda_2 t})$$

where:

$$(1) p + q = 1$$

$$(2) CV = \frac{1}{\sqrt{p}}$$

(3)  $\lambda$  = Rate is specified and then

$\lambda_1, \lambda_2$  chosen to satisfy--

$$\lambda_2 = (2 + \sqrt{2})\lambda$$



$$\lambda_1 = \left( \frac{p(2 + \sqrt{2})}{2 + \sqrt{2} - q} \right) \lambda$$

Input parameters are:

- (1) Rate, specified in terms of mean number of events per minute.
- (2) Coefficient of variation (must be greater than 1.0)

c. Degenerate (Fixed Rate)

$$F(t) = \begin{cases} 0 & t < 1/\lambda \\ 1 & t \geq 1/\lambda \end{cases}$$

Input parameter:

- (1) Rate  $\lambda$  specified in terms of mean number of events per minute.

## 2. Queuing Disciplines

For comparison purposes, there are four queuing disciplines to choose from. These determine the priority by which members of the queue will enter service.

a. FIFO - the first into queue is the first out. Over all disciplines which are independent of service times, this one minimizes the variance of delay.

b. LIFO - the last into the queue is the first out. Some systems, by their nature, function in this manner.

c. SSTF - the queue member with shortest service time is served first; this minimizes the mean delay over



all other disciplines, but the variance of delay is greater than that of FIFO.

d. RANDOM DRAW - applies to a system in which insufficient information is known to implement any other discipline.

### 3. Length of the Simulation

The period of simulation is specified in minutes by the user. All queues are empty and servers idle at the start of the simulation.



### III. THE METHOD

#### A. THE COMPUTER SIMULATION

The computer simulation is a modified next event type. There are two types of events, arrivals to the queue and a completion of service. With the first there will be an associated entry to service if there is an idle server, and with the last an associated entry to service if the queue is not empty.

The simulation is begun with the generation of the first arrival time and its associated service time. The times are generated via the transformation of random numbers to stochastic variates of the type specified. At the occurrence of an arrival, the next arrival time and associated service time are generated. When a server is available, a waiting member of the queue enters service with completion time equal to the sum of his assigned service time and the time he departs the queue. Thus with the clock starting at zero, a time history of events is created which tells what happened and when.

At the same time, the appropriate values are retained from which statistics of the simulation can later be calculated.

#### B. THE GRAPHICAL DISPLAY

At the completion of the simulation phase, the time history is depicted dynamically on the cathode ray tube



in real time. Pictures 1 to 5 show the form of the display, with the man-like images in motion along the appropriate path, and the number in the queue box showing the queue's current length.

The timing and motion of the display are controlled by the analog computer via the hybrid connections to the digital computer. The first connection is for the purpose of timing until the next event is to be initiated. In this case, the analog computer integrates from zero to a preset time value in volts, at the rate of one volt per second. The interrupt to the appropriate subroutine then occurs, and the analog computer is reset with the next time value, after which the integration is reinitiated.

The second connection causes the dynamic motion to occur by continually checking events in progress and causing the position of the corresponding images to be incremented appropriately; this interrogation and incrementation occurs approximately 10 times per second depending on the settings of the analog computer. Movement increments are between one-half inch and two inches (depending on the path), causing a motion of between five inches and 20 inches per second.



#### IV. STATISTICS FROM THE SIMULATION

STATISTICS					
	D1	D2		D1	D2
DT	.58	3.98	DC	.31	3.67
BT	2.31	.95	BC	2.27	2.33
LT	2.88	4.94	LC	2.58	6.88
DC	1.15	22.24	DV	5.65	111.86
SC	5.32	2.48	SV	26.59	11.27
WC	6.47	24.63	OV	3.67	38.14
DM	5.88	57.88	OM	3.88	9.88
D8	.58	.92	OS	.35	.81
DX	.46	.92	OL	.14	.76
NC	26	24	PL	.88	
ENTER 1. CONTINUE					
2. PARAMETER LISTING					

At the end of the simulation period, the following statistics are calculated and displayed, as shown above.

AT = time - average length of the queue

BT = time - average number of busy servers

LT = time - average number in the system = QT + BT

QC = customer - average length of the queue

BC = customer - average number of busy servers

LC = customer - average number in the system = QC + BC

DC = customer - average delay in the queue

SC = customer - average service time

WC = customer - average wait in the system = DC + SC



DM = maximum delay experienced by any customer

DO = probability that a customer's delay is greater than zero.

DX = probability that a customer's delay is greater than XD. (The value of XD is an input option.)

QM = maximum queue length reached during the simulation

QO = probability that the queue length will be greater than zero.

QL = probability that queue length will be greater than LQ (the value of LQ is an input option).

DV = standard deviation of the delay in queue

SV = standard deviation of the service times

QV = standard deviation of the queue length

NC = total number of arrivals

PS = percentage not lost (losses model only)



## V. PARAMETER INPUT

	ITYPE	5	RUNTIME	1.88
	IC	3	IC1	1
	1	2	3	4
IDIST	1	1	1	1
KK	1	1	1	1
RATE	38.88	38.88	14.88	14.88
CV	1.88	1.88	1.88	1.88
IDSPLN	1	1		
	LD	1	MAXD	1
	XD	1.88	NPOP	1
	IDUAL	1	P	1.88
ENTER 1. CONTINUE				
2. STATISTICS				

Picture 7

Parameters can be input by the card reader or by the teletype terminal. In either case, the namelist form is used, where the entry form is NAME = X, NAME being the variable title and X being the value being assigned. For real variables, the decimal point must be entered. Namelist input is terminated with a \*.

### A. SAMPLE INPUT CARD

---

```
IDEV = 2, ITYPE = 3, RATE(2) = 3.0 *
```



## B. NAMELIST INPUT VIA THE GRAPHICS TERMINAL

If this mode of entry is specified, one variable at a time is entered, followed by a carriage return. Entries are displayed, as entered, on the graphics screen. Termination occurs with the entry of a \* followed by a carriage return.

## C. INPUT VARIABLES

The following conventions are employed for specifying the options and parameters to be used.

ITYPE = 1 gives G/G/C

2 gives G/G/C with losses

3 gives G/G/C with finite source

4 gives G/G/C with feedback

5 gives two simultaneous G/G/C queues

RUNTIME = desired length of simulation (minutes)

IC = number of servers for queue 1

IC1 = number of servers for queue 2 (only applicable if  
ITYPE = 5 is chosen)

IDIST(I) = 1. K - ERLANG

2. HYPEREXPONENTIAL

3. DEGENERATE

KK(I) = K in K-ERLANG (use K=1 for exponential)

RATE(I) = Rate in number per minute

CV(I) = Coefficient of variation (specify for IDIST(I) = 2  
only. This variable is computed otherwise.)



where I = 1 for arrivals to queue 1 (except finite model)  
2 for arrivals to queue 2 or finite population  
idle parameters  
3 for service for queue 1 (except finite model)  
4 for service for queue 2 or finite population  
service parameters

IDSPLN(I) = 1. FIFO  
2. LIFO  
3. Shortest service time first  
4. Random draw

where I = 1 for queue 1  
2 for queue 2

LQ = parameter in the statistic  $P(Q > LQ)$

XD = parameter in the statistic  $P(\text{wait} > XD)$

IDUAL = 0. Independent arrival streams

1. Identical arrival streams  
(applicable to ITYPE = 5 only)

MAXQ = maximum queue size (for G/G/C with losses only)

NPOP = population size (for finite source option only)

P = completion probability (for feedback option only)



## VI. THE PROGRAM

The program requires the use of the XDS 9300 digital computer, the CI 5000 analog computer, and a graphics terminal using the GATED monitor.

The analog computer wiring diagram necessary to drive the display sequence appears in APPENDIX A. Before beginning, set Delay Flop Switches DF00, DF01, and DF02 to .1 MS and set counter F00 to 1 and F04 to 0. Then select RUN and DIGITAL COMPUTER on the analog control panel. This provides for movement interrupts to occur at a rate of 100/11 hertz. If desired, the timing values will be displayed on the RATIOMETER by selecting TRUNK 420 prior to DIGITAL COMPUTER.

The program consists of a FORTRAN program and a META-SYMBOL 9300 program. Because the history is buffered out to the device assigned the logical unit value 7, an additional control card is needed.

Job cards for deck run should be:

⌵JOB

⌵AGT

⌵ASSIGN 7=DF1A

⌵FORTRAN LS,GO

FORTRAN DECK

⌵META 9300 SI,LO,GO

META-SYMBOL DECK



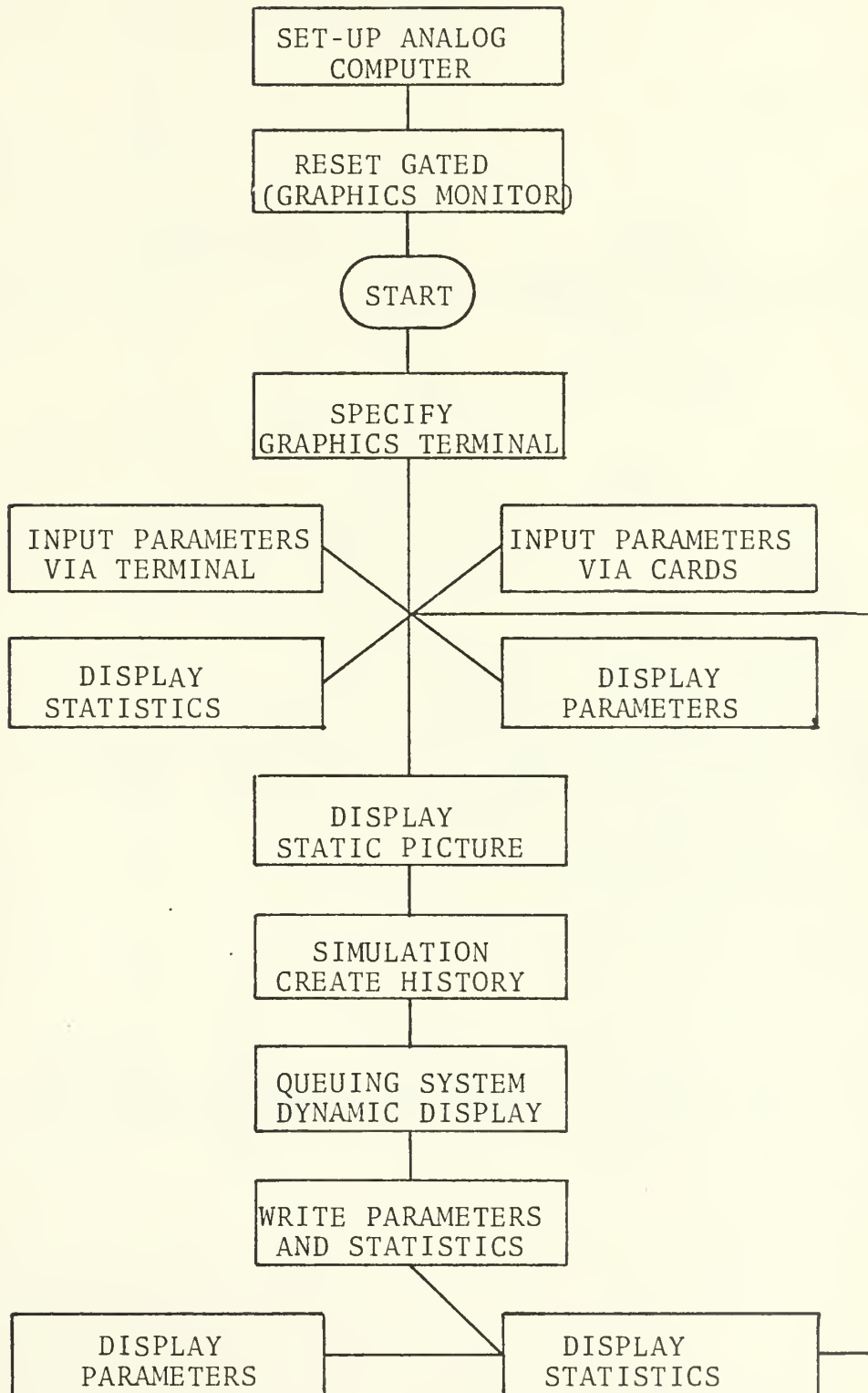
¬EOF

¬LOAD XR,MAP

¬DATA

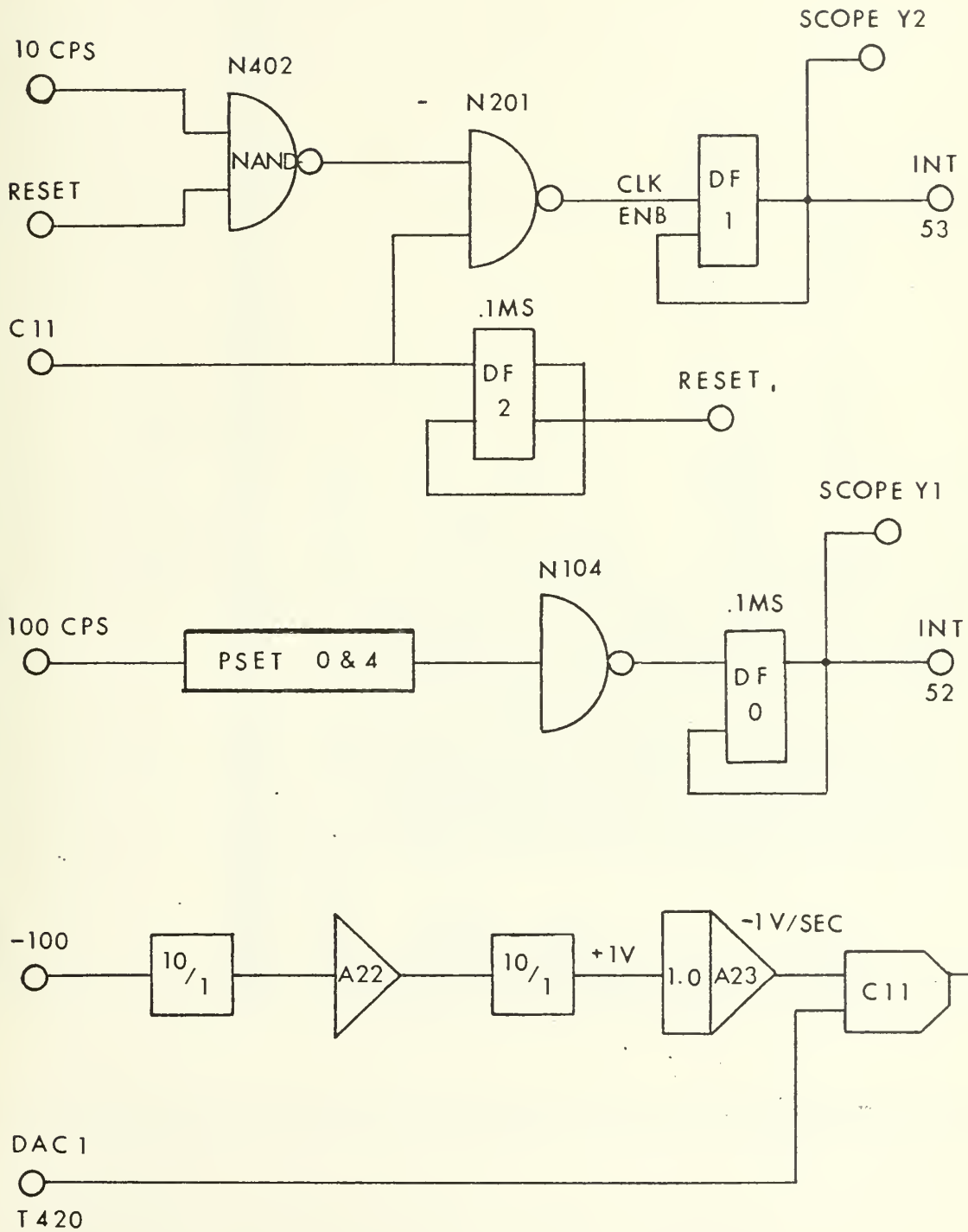


# PROGRAM STEPS





# APPENDIX A ANALOG SCHEMATIC





# SIMULATION OF QUEUEING MODELS WITH DYNAMIC DISPLAY

C. M. RIDDELL

MARCH 1973

## DICTIONARY OF EXOGENOUS VARIABLES

CHNG	-	INCREMENT VALUES	
CLOCK	-	HISTORY TIMES	
CLTIME	-	SERVICE TIME SUM	
CLTIMESQ	-	(SERVICE TIME) <sup>2</sup> SUM	
CTIMSQ	-	(QUEUE LENGTH) <sup>2</sup> SUM	
CUSTIM	-	QUEUE LENGTH SUM	
CV	-	COEFFICIENT OF VARIATION	
DELAY	-	QUEUE WAIT SUM	
DELAYO	-	DELAYS GREATER THAN 0	
DELAYSQ	-	(QUEUE WAIT) <sup>2</sup> SUM	
DELAYX	-	DELAYS GREATER THAN XD	
ENDD	-	MOVEMENT END VALUES	
ETIME	-	NEXT EVENT TIMES	
I3	-	HISTORY POINTER	
IBUFF	-	HISTORY BUFFER	
IBUSY	-	BUSY? 0 - NO; 1 - YES; 2 - NOT APPLICABLE	
IC	-	NUMBER OF SERVERS QUEUE 1	
IC1	-	NUMBER OF SERVERS QUEUE 2	
IDEV	-	SPECIFIES GRAPHICS TERMINAL	
IDIST	-	DISTRIBUTION TYPE	
IDSPLN	-	QUEUE DISCIPLINE	
IDUAL	-	IDENTICAL ARRIVALS? 0 - NO; 1 - YES	
IFDBK	-	FEEDBACK? 0 - NO; 1 - YES	
IFLAG	-	DYNAMIC IMAGE FLAGS	
ILN	-	TEXT LINE NUMBER	
IMAN	-	DYNAMIC IMAGES	
IMD	-	MOVE-DRAW INDICATORS	
IPOS	-	TEXT STARTING POSITION	
ISTDT	-	ARRIVAL BUSY SERVER SUM	



ITOT	INTERMEDIATE HISTORY COUNTER
ITPOP	POPULATION SIZE TEXT WORD
ITQ	QUEUE LENGTH TEXT WORDS
ITXA	MAIN CHOICE PAGE TEXT WORD
ITYPE	MODEL TYPE
IX1-IX4	RANDOM NUMBER SEEDS
KIM	HISTORY IMAGE POINTER
KK	K (ERLANG)
KP	VECTOR POINTERS FOR TIM
LINE	LENGTH OF QUEUES
LOG	HISTORY LOGIC POINTER
LQ	LOWER LIMIT QUEUE LENGTH VALUE
LTOT	ARRIVAL QUEUE SUM
MAXLN	MAXIMUM QUEUE VALUE REACHED
MAXO	DESIRED MAX QUEUE LENGTH
MD	MOVE-DRAW INDICATORS
XMDELAY	MAXIMUM DELAY IN QUEUE
NCUST	ARRIVAL COUNTERS
NFOBK	TOTAL NUMBER OF FEEDBACKS
NPPOP	FINITE POPULATION SIZE
NQ	LENGTH OF QUEUES
NSIM	NUMBER OF STATIC IMAGES
NTOT	TOTAL NUMBER OF HISTORY ENTRIES
NULL	BLANK
P	PROBABILITY
PROBO	QUEUE GREATER THAN 0 SUM
PROBL	QUEUE GREATER THAN LQ SUM
PSPTIME	PREVIOUS STATE CHANGE TIME
PPTIME	PREVIOUS QUEUE CHANGE TIME
RATE	RATE
RUNTIME	LENGTH OF SIMULATION
TIM	STORED TIMES
TIME	CURRENT TIME
XD	LOWER LIMIT TIME VALUE
X	SCREEN X COORDINATES
XX	DYNAMIC IMAGE X COORDINATES
XXX	DYNAMIC IMAGE STARTING POSITIONS
YYY	SCREEN Y COORDINATES
YY	DYNAMIC IMAGE Y COORDINATES
YY	DYNAMIC IMAGE STARTING POSITIONS







```

C 52 INTERRUPT CAUSES MOTION
C 53 INTERRUPT TIMES EVENTS
C
C CONNECT (52,FLAGS)
C CONNECT (53,SETFLAGS)
C
C DEFAULTS: NO HISTORY OUTPUT; GRAPHICS TERMINAL 1
C
C IPRINT=0; IDEV=1
C OUTPUT(101) TYPE IDEV=2 * AND A C/R IF AGT-2 IS TO BE USED ELSE T
C -TYPE * AND A C/R,
C INPUT(101)
C
C SET THE STATIC IMAGE COORDINATES
C
C CALL PACK
C
C INITIALIZE THE GRAPHICS AND TEXT DIRECTORIES
C
C CALL DGINIT (IDEV,IGDIR,30,IER)
C CALL DTINIT (IDEV,ITDIR,16,IER)
C
C 99 PRESENT MAIN CHOICE PAGE
C
C 110 ENCODE(24,101,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,15,25,2,3,IER)
C ENCODE(24,102,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,17,25,2,3,IER)
C ENCODE(24,104,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,19,25,2,3,IER)
C ENCODE(24,105,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,21,25,2,3,IER)
C ENCODE(24,103,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,24,25,2,3,IER)
C CALL TEXTR (IDEV,NULL,1,26,47,2,3,IER)
C
C WAIT FOR REPLY
C
C 100 IF(MOD(ITDIR(6),8).EQ.0) GO TO 100
C CALL TEXTI (IDEV,1,1,26,47,IER)
C
C LOGICAL RIGHT SHIFT 18 BITS (3 CHARACTERS) DECODES REPLY
C
C I=LRS(1,18)
C
C CALL DTINIT(IDEV,ITDIR,16,IER)
C
C GET CARD INPUT

```



```

C
      IF(I.NE.1) GO TO 113
      INPUT(5)
      DO 114 J=1,4
      IF(IDIST(J).EQ.3) CV(J)=0.0
      IF(IDIST(J).EQ.1) CV(J)=1.0/SQRT(KK(J))
114      GO TO 110
113      IF(I.NE.3) GO TO 112
      IF(I.NE.3) GO TO 112
111      CALL PARAMETERS
      CALL TEXTR (IDEV, NULL, 1, 34, 9, 2, 3, IER)
120      IF(MOD(ITDIR(14), 8).EQ.0) GO TO 120
      CALL TEXTI (IDEV, I, 1, 34, 9, IER)
      CALL DTINIT (IDEV, ITDIR, 16, IER)
      I=LRS(I, 18)
      IF(I.EQ.1) GO TO 110
      IF(I.NE.2) GO TO 111
      CALL STATISTICS
121      CALL TEXTR (IDEV, NULL, 1, 35, 11, 2, 3, IER)
125      IF(MOD(ITDIR(15), 8).EQ.0) GO TO 125
      CALL TEXTI (IDEV, I, 1, 35, 11, IER)
      CALL DTINIT (IDEV, ITDIR, 16, IER)
      I=LRS(I, 18)
      IF(I.EQ.1) GO TO 110
      IF(I.EQ.2) GO TO 111
      GO TO 121
112      IF(I.EQ.4) GO TO 1
      IF(I.NE.2) GO TO 110
      IRLK=1
C      GET TERMINAL INPUT
C
      CALL GINPUT (IDEV, ITDIR, IBLK)
      DO 115 J=1,4
      IF(IDIST(J).EQ.3) CV(J)=0.0
      IF(IDIST(J).EQ.1) CV(J)=1.0/SQRT(KK(J))
115      GO TO 99
C      CONVERT MINUTES TO SECONDS
C
      DO 2 I=1,4
      RATE(I)=RATE(I)/60.
      RUNTIME=RUNTIME*60
C
      BEGIN MODEL INITIALIZATION OF VALUES
C
      IF(ITYPE.GE.5) GO TO 4
      DO 3 I=1,6

```







```

21 ENDD(5,I)=X(14)+.1
   CHNG(1)=.225; CHNG(2)=.17; CHNG(3)=.205
   NSIM=6
   GO TO 85
C
C
C
   SETUP G/G/C WITH FINITE SOURCE
30 CALL FINITE
   A1=A2=7
   DO 31 I=1,4
31 ENDD(5,I)=X(9)+.03
   DO 32 I=5,10
32 ENDD(1,I)=X(12)
   DO 33 I=11,22
   ENDD(1,I)=X(13)+.1
   ENDD(2,I)=Y(24)-.1
   ENDD(3,I)=X(5)
   ENDD(4,I)=Y(16)
33 CHNG(1)=.12; CHNG(2)=.17; CHNG(3)=.2; CHNG(4)=.275
   NSIM=7
   GO TO 85
C
C
C
   SETUP G/G/C WITH FEEDBACK TO THE QUEUE
40 CALL FEEDBK
   DO 41 I=1,4
41 ENDD(5,I)=X(9)
   DO 42 I=5,10
42 ENDD(1,I)=X(12)
   DO 43 I=11,22
   ENDD(1,I)=X(13)+.1
   ENDD(2,I)=Y(24)-.1
   ENDD(3,I)=X(8)
   ENDD(4,I)=Y(12)
43 ENDD(5,I)=X(14)+.1
   CHNG(1)=.225; CHNG(2)=.17; CHNG(3)=.2; CHNG(4)=.35
   NSIM=7
   GO TO 85
C
C
C
   SETUP TWO G/G/C QUEUES
50 CALL TWOGGC
   ILN(1)=17; ILN(2)=27; IPOS(1)=IPOS(2)=41
   A1=A2=1; A3=5; A4=19
   G1=G2=10; G3=5; G4=19
   S1=S2=12; S3=5; S4=9; S5=2; S6=19; S7=22; S8=15
   DO 51 I=1,4
51 ENDD(5,I)=X(9)

```







```

CALL DTINIT(IDEV,ITDIR,16,IER)
I=LRS(I,18)
IF(I.EQ.1) GO TO 110
IF(I.NE.2) GO TO 91
C
C
C  DISPLAY PARAMETERS?
C
C  93  CALL PARAMETERS
C  175 CALL TEXTR(IDEV,NULL,1,34,9,2,3,IER)
      IF(MOD(ITDIR(14),8).EQ.0) GO TO 175
      CALL TEXTI(IDEV,I,1,34,9,IER)
      CALL DTINIT(IDEV,ITDIR,16,IER)
      I=LRS(I,18)
      IF(I.EQ.2) GO TO 91
      RETURN FOR ANOTHER RUN
C
C
C  GO TO 110
END

```



```

CCCCCCCCCCCCCCCC
SUBROUTINE SIMULATE
PURPOSE: TO CREATE THE TIME HISTORY OF EVENTS
          AND GATHER STATISTICS
SUBROUTINES USED: ARRIV, BEGSRV, ENDSRV, ACCUM, TIMES
CCCCCCCCCCCCCCCC

SUBROUTINE SIMULATE
DIMENSION Ibuff(400)
REAL L,LC
COMMON /SET6/ CLOCK(100),KIM(100),LOG(100)
COMMON /SET7/ ETIME(9),I,I3,K1,TIM(115,4)
COMMON /SET8/ CHNG(10),ENDD(5,22),IFLAG(22),ITOT,IPOS(2),XX(5,22),
-ILN(2),IMAN(7,22),NSIM,NQ(2),ITQ(2,2),NPOP,ITPOP(2),XX(5,22),
-XX(5,22),YY(5,22),YY(5,22),IMD(5)
COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IFDBK,IDSPLN(2),
-IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PSTIME,PTIME(2),RUNTIME,IX1,
-IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),DELAYSQ(2),XMDELAY(2),
-DELAYO(2),DELA YX(2),CLTIME(2),CLTIMSQ(2),CUSTIM(2),CTIMSQ(2),
-PROB0(2),PROBL(2),LTOT(2),ISTOT(2)
COMMON /STAT/ QT(2),BT(2),LT(2),QC(2),BC(2),LC(2),
-DC(2),SC(2),WC(2),DM(2),DO(2),DX(2),QM(2),QO(2),
-QL(2),DV(2),SV(2),OV(2),PL,NC(2)
EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
- (IBUFF(301),LOG(1))
I3=1
I=2
SET EVENT TIMES TO LARGE VALUE
DO 10 K=1,8
10 ETIME(K)=99999.
IF(ITYPE.NE.3) GO TO 4
I=2
IDLE TIMES FOR FINITE POPULATION
DO 3 J=1,NPOP
3 CALL TIMES
TIM(KP(1)+1,1)=TIM(KP(2),2)
TIM(KP(1)+1,3)=TIM(KP(2),4)

```



```

C      KP(2)=KP(2)-1
C      SET FIRST ARRIVAL FROM FINITE SOURCE
C
C      ETIME(7)=TIM(KP(1)+1,1)
C      GO TO 5
C
C      GET FIRST ARRIVAL TIME
C
C      4 I=1; CALL TIMES; ETIME(7)=TIM(1,1)
C        IF(ITYPE.EQ.5) I=2; CALL TIMES; ETIME(8)=TIM(1,2)
C
C      SET LENGTH OF SIMULATION
C      5 ETIME(9)=RUNTIME
C
C      DETERMINE NEXT EVENT
C
C      11 TIME=ETIME(1)
C        K1=1
C        DO 15 K=2,9
C          IF(ETIME(K).LT.TIME) TIME=ETIME(K); K1=K
C        15 CONTINUE
C
C      SEPARATE END-OF-SERVICE
C
C      IF(K1.GT.6) GO TO 20
C      I=1
C      IF(ITYPE.GE.5 .AND. K1.GE.4) I=2
C      CALL ENDSRV; GO TO 11
C
C      SEPARATE ARRIVALS
C
C      20 IF(K1.LE.8) I=K1-6; CALL ARRIV; GO TO 11
C
C      DONE: COMPLETE HISTORY
C
C      CALL BUFFEROUT(7,1,IBUFF,400,IND)
C      21 IF(IND.EQ.1) GO TO 21
C      GO TO (21,23,22,22) IND
C      22 OUTPUT(101) 'BUFFERING ERROR 5'
C      23 REWIND 7
C      I=1
C      {
C      COMPLETE TIME STATISTICS
C
C      50 CALL ACCUM
C

```



C

# CALCULATE FINAL STATISTICAL VALUES

```

TCUST=NCUST(I)
IF(ITYPE.EQ.2) TCUST=NCUST(1)+NCUST(2);
-PL=NCUST(1)/TCUST; NCUST(1)=TCUST
DO 55 J=1,KP(I)
  WAIT=TIME-TIM(J,I)
  DELAY(I)=DELAY(I)+WAIT
  IF(WAIT.GT.XMDELAY(I)) XMDELAY(I)=WAIT
  DELAYO(I)=DELAYO(I)+1
55 IF(WAIT.GT.XD) DELAYX(I)=DELAYX(I)+1
  SC(I)=CLTIME(I)/(TCUST-LINE(I))
  SV(I)=SORT(CLTIMSQ(I)/(TCUST-LINE(I))-SC(I)**2)
  K=1; M1=1
  IF(ITYPE.NE.5) M2=6; GO TO 59
  IF(I.EQ.1) M2=3; GO TO 59
  M1=4; M2=6; K=2
59 DO 60 J=M1,M2
  IF(ETIME(J).EQ.99999.) GO TO 60
  CLTIME(K)=CLTIME(K)-(ETIME(J)-TIME)
60 CONTINUE
  QT(I)=CUSTIM(I)/TIME
  BT(I)=CLTIME(I)/TIME
  LT(I)=QT(I)+BT(I)
  QC(I)=LT(I)/TCUST
  BC(I)=ISTOT(I)/TCUST
  LC(I)=QC(I)+BC(I)
  DC(I)=DELAY(I)/TCUST
  WC(I)=DC(I)+SC(I)
  DM(I)=XMDELAY(I)
  DO(I)=DELAYO(I)/TCUST
  DX(I)=DELAYX(I)/TCUST
  QM(I)=MAXLN(I)
  QO(I)=PROB(I)/TIME
  QL(I)=PROB(I)/TIME
  DV(I)=SORT(DELAYSQ(I)/TCUST-DC(I)**2)
  QV(I)=SORT(CTIMSQ(I)/TIME-QT(I)**2)
  NC(I)=NCUST(I)
  IF(ITYPE.NE.2) PL=0.0
  IF(ITYPE.NE.5) GO TO 70
  IF(I.EQ.1) I=2; GO TO 50
  GO TO 80
70 QT(2)=BT(2)=LT(2)=QC(2)=BC(2)=LC(2)=DC(2)=SC(2)=
  -WC(2)=DV(2)=SV(2)=QV(2)=DM(2)=DO(2)=DX(2)=QM(2)=
  -QO(2)=QL(2)=NC(2)=0
80 RETURN
END

```



SUBROUTINE ARRIV

PURPOSE: RECORD AN ARRIVAL TO THE QUEUE IN THE HISTORY,  
GATHER STATISTICS, AND DETERMINE IF A SERVER  
IS AVAILABLE

此後必能與世同休矣

```

SUBROUTINE ARRIV
DIMENSION Ibuff(400)
DIMENSION NO(2)
DATA NO/1,2/
COMMON /SET6/ CLOCk(100),KIM(100),LOG(100)
COMMON /SET7/ FTIME(9),I,I3,K1,TIM(115,4)
COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFD BK,IFDBK,IDSPLN(2),
- IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PSTIME,PTIME,IX1,
- IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),XMDelay(2),
- DELAYO(2),DELAyx(2),CLTIME(2),CLTIMSQ(2),CUSTIM(2),CTIMSQ(2),
- PROBO(2),PROBL(2),LTOT(2),ISTOT(2)
EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
- (IBUFF(301),LOG(1))
KP(1)=KP(1)+1

```

```

CCCC
      QUEUE AT CAPACITY? STOP
      IF(KP(1).GE.114) ETIME(9)=TIME-1.0; RETURN
CCCC
      GATHER STATISTICS

```

```

50 LTOT(I)=LTOT(I)+LINE(I)
   IF(ITYPE.EQ.5) GO TO 51
   DO 50 J=1,6
   IF(IBUSY(J).EQ.2) GO TO 54
   ISTOT(I)=ISTOT(I)+IBUSY(J)
   GO TO 54
51 DO 52 J=1,3
   IF(IBUSY(J).EQ.2) GO TO 54
   ISTOT(I)=ISTOT(I)+IBUSY(J)
52 IF(IFDBK.EQ.1) GO TO 13
54 IF(ITYPE.NE.2) GO TO 5

```

CC  
IN (11) PLANE 27, 000  
DETERMINE IF LOST



```

C      IF(LINE(1).LT.MAXQ) GO TO 5
C
C      HISTORY ENTRY
C
C      CLOCK(I3)=TIME-PSTIME
C      KIM(I3)=NO(2)
C
C      ALTERNATE IMAGES
C
C      IF(NO(2).EQ.2) NO(2)=4: GO TO 101
C      NO(2)=2
C      LOG(I3)=2
C      I3=I3+1: NTOI=NTOI+1
C      IF(I3.NE.101) GO TO 4
C      CALL BUFFEROUT(7,1,IBUFF,400,IND)
C      IF(IND.EQ.1) GO TO 1
C      GO TO (1,3,2,2) IND
C      OUTPUT(101)'BUFFERING ERROR 1'
C      I3=1
C      NCUST(2)=NCUST(2)+1
C      PSTIME=TIME
C      KP(1)=KP(1)-1
C      GO TO 40
C      IF(ITYPE.GE.5 .OR. ITYPE.EQ.2) J=1: GO TO 6
C      IF(J.EQ.1) J=2: GO TO 6
C      J=1
C
C      ALTERNATE IMAGES
C
C      IF(NO(J).EQ.J) NO(J)=J+2: GO TO 8
C      NO(J)=J
C
C      HISTORY ENTRY
C
C      KIM(I3)=NO(J)
C      CLOCK(I3)=TIME-PSTIME
C      LOG(I3)=1
C      I3=I3+1: NTOI=NTOI+1
C      IF(I3.NE.101) GO TO 12
C      CALL BUFFEROUT(7,1,IBUFF,400,IND)
C      IF(IND.EQ.1) GO TO 9
C      GO TO (9,11,10,10) IND
C      OUTPUT(101)'BUFFERING ERROR 2'
C      I3=1
C
C      TIME STATISTICS
C

```



```

13 NCUST(I)=NCUST(I)+1
   LINE(I)=LINE(I)+1
   IF(IFDBK.EQ.1) IFDBK=0: GO TO 42
C
C
C
   SERVER AVAILABLE?
   IF(LINE(I).GT.1) GO TO 40
   LOG(I3)=7
   IF(ITYPE.LE.4) GO TO 30
   IF(I.EQ.2) GO TO 20
   DO 15 K1=1,3
15  IF(IBUSY(K1).EQ.0) CALL BEGSRV: GO TO 40
   GO TO 40
20  DO 21 K1=4,6
21  IF(IBUSY(K1).EQ.0) CALL BEGSRV: GO TO 40
   GO TO 40
30  DO 31 K1=1,6
31  IF(IBUSY(K1).EQ.0) CALL BEGSRV: GO TO 40
40  IF(ITYPE.NE.3) GO TO 45
C
C
C
   FINITE SOURCE EMPTY?
   IF(KP(2).EQ.0) ETIME(7)=99999.: GO TO 42
C
C
C
   SWITCH NEXT ARRIVAL TO QUEUE HALF OF ARRAY
   TIM(KP(1)+1,1)=TIM(KP(2),2)
   TIM(KP(1)+1,3)=TIM(KP(2),4)
   KP(2)=KP(2)-1
   GO TO 41
C
C
C
   GET NEXT ARRIVAL TIMES
45  CALL TIMES
41  ETIME(I+6)=TIM(KP(I)+1,I)
42  IF(LINE(I).GT.MAXLN(I)) MAXLN(I)=LINE(I)
   RETURN
   END

```



```

C
C
C
C
C
C
C
SUBROUTINE BEGSRV
PURPOSE:  RECORD BEGINNING OF SERVICE;  SELECTS QUEUE MEMBER
AND SETS END OF SERVICE TIME

C
C
C
C
C
C
SUBROUTINE BEGSRV
DIMENSION Ibuff(400)
COMMON /SET6/ CLOCK(100),KIM(100),LOG(100)
COMMON /SET7/ ETIME(9),I,I3,K1,TIM(115,4)
COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IFDBK,IDSPLN(2),
- IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PSTIME,PTIME(2),RUNTIME,IX1,
- IX2,IX3,IX4,XD,LQ,MAXQ,PE
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),DELAYSQ(2),XMDELAY(2),
- DELAYO(2),DELAYX(2),CLTIME(2),CLTIMSQ(2),CUSTIM(2),CTIMSQ(2),
- PROBO(2),PROBL(2),LTOT(2),ISTOT(2)
EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
- (IBUFF(301),LOG(1))
HISTORY ENTRY
CLOCK(I3)=TIME-PSTIME
KIM(I3)=K1+4
I3=I3+1; NTOT=NTOT+1
IF(I3.NE.101) GO TO 4
CALL BUFFEROUT(7,1,IBUFF,400,IND)
1 IF(IND.EQ.1) GO TO 1
GO TO (1,3,2,2) IND
2 OUTPUT(101),BUFFERING ERROR 4,
3 I3=1
SERVER BUSY
4 IBUSY(K1)=1
II=I+2
SELECT QUEUE DISCIPLINE
GO TO (10,20,30,40) IDSPLN(I)
FIFO

```



```

10 STIME=TIM(1,II)
   WAIT=TIME-TIM(1,I)
   K=1
   GO TO 50
C
C
C
   LIFO
20 STIME=TIM(KP(I),II)
   WAIT=TIME-TIM(KP(I),I)
   K=KP(I)
   GO TO 50
C
C
C
   SSTF
30 STIME=TIM(1,II)
   K=1
   DO 35 J=2,KP(I)
   IF(TIM(J,II).LT.STIME) STIME=TIM(J,II); K=J
   WAIT=TIME-TIM(K,I)
   GO TO 50
C
C
C
   RANDOM
40 PIECE=1.0/KP(I)
   IX4=IX4+4099; RN=0.5+IX4*.5960464E-7
   K=1
   REPEAT 45 WHILE (K*PIECE).LT.RN
45 K=K+1
   STIME=TIM(K,II)
   WAIT=TIME-TIM(K,I)
C
C
C
   ADJUST ARRAY
50 DO 55 J=K,KP(I)
   TIM(J,I)=TIM(J+1,I)
55 TIM(J,I)=TIM(J+1,II)
   KP(I)=KP(I)-1
C
C
C
   MINIMUM SERVICE TIME 1.2 SECONDS (TIME FOR MOTION)
   STIME=AMAX(STIME,1.2)
C
C
C
   STATISTICS
60 DELAY(I)=DELAY(I)+WAIT
   DELAYSQ(I)=DELAYSQ(I)+WAIT**2
   IF(WAIT.GT.XMDELAY(I)) XMDELAY(I)=WAIT
   IF(WAIT.GT.0.0) DELAYO(I)=DELAYO(I)+1
C
C
C

```



```

IF(WAIT.GT.XD) DELAYX(I)=DELAYX(I)+1
CLTIME(I)=CLTIME(I)+STIME
CLTIMSQ(I)=CLTIMSQ(I)+STIME**2
SET END-OF-SERVICE-TIME
ETIME(K1)=TIME+STIME
LINE(I)=LINE(I)-1
RETURN
END

```

```

C
C
C

```



```

CCCCCCCCCCCC
SUBROUTINE ENDSRV
PURPOSE:  RECORD END-OF-SERVICE IN HISTORY AND DETERMINE
          IF BEGIN SERVICE FOLLOWS
*****
SUBROUTINE ENDSRV
DIMENSION IBUFF(400)
DIMENSION NO(6)
DATA NO/6*0/
COMMON /SET6/  CLOCK(100),KIM(100),LOG(100)
COMMON /SET7/  ETIME(9),I,I3,K1,TIM(115,4)
COMMON /SET9/  IDIST(4),RATE(4),KK(4),CV(4),NFD BK,IFDBK,IDSPLN(2),
- IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PSTIME,PRTIME,IX1,
- IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),DELA YSQ(2),XMDELAY(2),
- DELAYO(2),DELA YX(2),CLTIME(2),CLTIMSQ(2),CUSTIM(2),CTIMSQ(2),
- PROBO(2),PROBL(2),LTOT(2),ISTOT(2)
- EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
- (IBUFF(301),LOG(1))
SERVER NOT BUSY
IBUSY(K1)=0
IF(ITYPE.NE.4) GO TO 10
DETERMINE IF FEEDBACK
IX3=IX3+4099; RN=0.5+IX3*.5960464E-7
IF(RN.GT.P) IFDBK=1; NFD BK=NFD BK+1
HISTORY ENTRY
10 CLOCK(I3)=TIME-PSTIME
ALTERNATE IMAGES
IF(NO(K1).EQ.0) NO(K1)=1; GO TO 100
NO(K1)=0
KIM(I3)=K1+10+6*NO(K1)
IF(IFDBK.EQ.1) LCG(I3)=2; GO TO 11
IF(ITYPE.EQ.3) LOG(I3)=2; GO TO 11
100
*****

```











```

CCCCCCCCCCCC
SUBROUTINE TIMES
PURPOSE: TO GENERATE AND STORE ARRIVAL (OR IDLE) TIMES
          AND SERVICE TIMES

SUBROUTINE TIMES
COMMON /SET7/ ETIME(9), I, I3, K1, TIM(115,4)
COMMON /SET9/ IDIST(4), RATE(4), KK(4), CV(4), NFDBK, IFDBK, IDSPLN(2),
- IDUAL, IBUSY(6), KP(2), LINE(2), TIME, PSTIME, PTIME(2), RUNTIME, IX1,
- IX2, IX3, IX4, XD, LQ, MAXQ, P
COMMON /SET10/ NTOT, ITYPE
COMMON /SET11/ NCUST(2), MAXLN(2), DELAY(2), DELAYSQ(2), XMDELAY(2),
- DELAYO(2), DELAYX(2), CLTIME(2), CLTIMSQ(2), CUSTIM(2), CTIMSQ(2),
- PROBO(2), PROBL(2), LIOT(2), ISTOT(2)

FIRST TIME THROUGH GETS ARRIVAL (OR IDLE) TIME
IDLE TIMES STORED IN COLUMN 2
I2=1
FEEDBACK? SLIP INTO QUEUE TIME ARRAY
IF( IFDBK.EQ.1) TIM(KP(1)+2, I2)=TIM(KP(1)+1, I2);
- TIM(KP(1)+2, I2+2)=TIM(KP(1)+1, I2+2);
- TIM(KP(1)+1, I2)=TIME; I2=I+2; GO TO 1
IF( ITYPE.NE.5) GO TO 1

DUAL ARRIVAL STREAMS?
IF( IDUAL.EQ.1.AND.I.EQ.2) TIM(KP(2)+1,2)=TIM(KP(1)+1,1); I2=I+2

SELECT DISTRIBUTION
1 GO TO (10,20,30) IDIST(I2)
K-ERLANG
10 U=1.0/(KK(I2)*RATE(I2))
SV=0.0
DO 11 J=1, KK(I2)
C GENERATE RANDOM NUMBER BETWEEN 0 AND 1

```



```

C
11 IX1=IX1*4099; RN=0.5+IX1*.5960464E-7
   SV=SV-U*ALOG(RN)
   GO TO 88
C
C
C   HYPEREXPONENTIAL
20 IX1=IX1*4099; RN1=0.5+IX1*.5960464E-7
   IX2=IX2*4099; RN2=0.5+IX2*.5960464E-7
   U=1.0/RATE(I2)
   P=1.0/(CV(I2)**2)
   UX=0.2928932
   IF(P*GE*RN1) UX=(U-((1.0-P)*UX))/P
   SV=-UX*ALOG(RN2)
   GO TO 88
C
C
C   DEGENERATE
30 SV=1.0/RATE(I2)
88 IF(I2.NE.1) GO TO 89
C
C
C   MINIMUM INTER-ARRIVAL TIME=0.5 ( MOTION TIME)
-GO TO 87
IF(ITYPE.EQ.2 .OR. ITYPE.EQ.5) SV=AMAX(SV,0.5);
C
C
C   MINIMUM INTER-ARRIVAL TIME=0.3 (MOTION TIME)
SV=AMAX(SV,0.3)
87 TIM(KP(I)+1,I2)=SV+TIME
   I2=I+2
   GO TO 1
89 TIM(KP(I)+1,I2)=SV
   IF(ITYPE.NE.3) GO TO 95
   IF(KP(2).EQ.0) GO TO 91
C
C
C   NEXT END-IDLE TIME AT TAIL OF ARRAY
TEMP1=TIM(KP(I)+1,2)
TEMP2=TIM(KP(I)+1,4)
J=KP(I)
REPEAT 90 WHILE TEMP1.GT.TIM(J,2) .AND. J.GT.0
   TIM(J+1,2)=TIM(J,2); TIM(J+1,4)=TIM(J,4)
   J=J+1
90 TIM(J+1,2)=TEMP1; TIM(J+1,4)=TEMP2
91 KP(2)=KP(2)+1
95 RETURN
   END

```



此其於世也。如松柏之長壽。如金石之堅固。如日月之光輝。如天地之大德。如江河之流注。如雲霞之舒卷。如風雨之潤澤。如雷霆之震怒。如鬼神之神妙。如星辰之昭著。如山川之壯麗。如草木之花實。如鳥獸之音聲。如蟲魚之鱗介。如百物之生靈。如萬民之衆庶。如宇宙之大全。如造化之玄機。如道法之真諦。如神仙之境界。如佛祖之涅槃。如儒教之仁義。如墨家之兼愛。如法家之法術。如兵家之謀略。如農家之稼穡。如工家之器用。如醫家之藥石。如商賈之貨財。如士大夫之禮儀。如婦人女子之節操。如童蒙之學問。如老幼之孝悌。如貧富之安樂。如貴賤之分別。如名譽之顯赫。如罪惡之深重。如福祿之豐盈。如災祥之變幻。如生死之循環。如因果之報應。如夢幻之泡影。如虛空之無常。如一切之皆然。如一切之皆非。如一切之皆是。如一切之皆否。如一切之皆然。如一切之皆非。如一切之皆是。如一切之皆否。

## PURPOSE: DISPLAY THE STATISTICS PAGE

卷之五

52



```

ENCODE(44, 9, IWORD) DM(1), DM(2), QM(1), QM(2)
CALL TEXT0 (IDEV, IWORD, 11, 25, 7, 2, 3, IER)
ENCODE(44, 10, IWORD) DO(1), DO(2), QO(1), QO(2)
CALL TEXT0 (IDEV, IWORD, 11, 27, 7, 2, 3, IER)
ENCODE(44, 11, IWORD) DX(1), DX(2), QL(1), QL(2)
CALL TEXT0 (IDEV, IWORD, 11, 29, 7, 2, 3, IER)
ENCODE(44, 12, IWORD) NC(1), NC(2), PL
CALL TEXT0 (IDEV, IWORD, 11, 32, 7, 2, 3, IER)
ENCODE(44, 13, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 34, 7, 2, 3, IER)
ENCODE(44, 14, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 36, 7, 2, 3, IER)
RETURN
END

```



CCCCCCCC

# SUBROUTINE PARAMETERS

PURPOSE: DISPLAY THE CURRENT PARAMETERS

```

SUBROUTINE PARAMETERS
DIMENSION IWORD(11)
COMMON /SET3/X(14), Y(25), MD(6), IC, IC1, NULL
COMMON /SET5/ IDEV, IER
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOP(2), XX(5,22),
-ILN(2), IMAN(7,22), NSIM, NQ(2), ITQ(2,2), NPOP, ITPOP(2), XX(5,22),
-XXX(5,22), YY(5,22), YYY(5,22), IMD(5)
COMMON /SET9/ IDIST(4), KK(4), CV(4), NFDBK, IFDBK, IDSPLN(2),
-IDUAL, IBUSY(6), KP(2), LINE(2), TIME, PTIME(2), RUNTIME, IX1,
-IX2, IX3, IX4, XD, LQ, MAXQ, P
COMMON /SET10/ NTOT, ITYPE
FORMAT(7X, 'ITYPE', 5X, I4, 6X, 'RUNTIME', F5.2, 3X)
FORMAT(7X, 'IC', 8X, I4, 6X, 'IC1', 7X, I4, 3X)
FORMAT(10X, 'I', 9X, '2', 9X, '3', 9X, '4', 3X)
FORMAT('KK', 14, 6X, I4, 6X, I4, 6X, I4, 3X)
FORMAT('RATE', 14, 6X, I4, 6X, I4, 6X, I4, 3X)
FORMAT('CV', F5.2, 5X, F5.2, 5X, F5.2, 5X, F5.2, 3X)
FORMAT('IDSPLN', 14, 6X, I4, 23X)
FORMAT(7X, 'LQ', 8X, I4, 6X, 'MAXQ', 6X, I4, 3X)
FORMAT(7X, 'XD', 7X, F5.2, 6X, 'NPOP', 5X, I4, 3X)
FORMAT(7X, 'IDUAL', 5X, I4, 6X, 'P', 8X, F5.2, 3X)
FORMAT(12X, 'ENTER 1. CONTINUE', 14X)
FORMAT(19X, '2. STATISTICS', 12X)
ENCODE(44, 1. IWORD) ITYPE, RUNTIME
CALL TEXT0(IDEV, IWORD, 11, 8, 7, 2, 3, IER)
ENCODE(44, 2. IWORD) IC, IC1
CALL TEXT0(IDEV, IWORD, 11, 10, 7, 2, 3, IER)
ENCODE(44, 3. IWORD)
CALL TEXT0(IDEV, IWORD, 11, 13, 7, 2, 3, IER)
ENCODE(44, 4. IWORD) IDIST(1), IDIST(2), IDIST(3), IDIST(4)
CALL TEXT0(IDEV, IWORD, 11, 15, 7, 2, 3, IER)
ENCODE(44, 5. IWORD) KK(1), KK(2), KK(3), KK(4)
CALL TEXT0(IDEV, IWORD, 11, 17, 7, 2, 3, IER)
ENCODE(44, 6. IWORD) RATE(1), RATE(2), RATE(3), RATE(4)
CALL TEXT0(IDEV, IWORD, 11, 19, 7, 2, 3, IER)
ENCODE(44, 7. IWORD) CV(1), CV(2), CV(3), CV(4)
CALL TEXT0(IDEV, IWORD, 11, 21, 7, 2, 3, IER)

```



```

ENCODE (44, 8, IWORD) IDSPLN(1), IDSPLN(2)
CALL TEXT0 (IDEV, IWORD, 11, 23, 7, 2, 3, IER)
ENCODE (44, 9, IWORD) LQ, MAXQ
CALL TEXT0 (IDEV, IWORD, 11, 26, 7, 2, 3, IER)
ENCODE (44, 10, IWORD) XD, NPOP
CALL TEXT0 (IDEV, IWORD, 11, 28, 7, 2, 3, IER)
ENCODE (44, 11, IWORD) IDUAL, P
CALL TEXT0 (IDEV, IWORD, 11, 30, 7, 2, 3, IER)
ENCODE (44, 12, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 33, 7, 2, 3, IER)
ENCODE (44, 13, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 35, 7, 2, 3, IER)
RETURN
END

```



```

CCCCCCCC
SUBROUTINE DRIVER
PURPOSE: TO CAUSE THE DYNAMIC DISPLAY OF THE TIME HISTORY
OTHER SUBROUTINES: SETFLAGS, FLAGS, REMAN
CCCCCCCC

SUBROUTINE DRIVER
DIMENSION Ibuff(400)
COMMON /SET6/ CLOCK(100),KIM(100),LOG(100)
COMMON /SET8/ CHNG(10),ENDD(5,22),IFLAG(22),ITOT,IPOS(2),
-ILN(2),IMAN(7,22),NSIM,NQ(2),ITQ(2,2),NPOP,ITPOP(2),XX(5,22),
-XX(5,22),YY(5,22),YY(5,22),IMD(5)
COMMON /SET10/ NTOT,ITYPE
COMMON /SET12/ IDONE,I3,IM
EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
- (IBUFF(301),LOG(1))
10 IDONE=1
CALL BUFFERIN (7,1,IBUFF,400,IND)
11 IF(IND.EQ.1) GO TO 11
GO TO (11,13,12,12) IND
12 OUTPUT(101),BUFFERIN ERROR ,
13 I3=1

DISALLOWS TOO SMALL A VALUE AFTER BUFFERIN
IF(CLOCK(I3).LE.0.1) CLOCK(I3)=.1
SET MAX (HYBRID LIMITATION)
IF(CLOCK(I3).GT.99.) CLOCK(I3)=99.
SET TIMER FOR NEXT EVENT
CALL DAC(1,CLOCK(I3)/100.)
CALL COMPUTE
CALL ENABLE
WAIT
CCCCCCCC

19 IF(IDONE.EQ.1) GO TO 19
CALL DISABLE
IF(IDONE.EQ.2) GO TO 10

```



30 REWIND 7: RETURN  
END



[illegible]

SUBROUTINE SETFLAGS

**PURPOSE:** SET IMAGE FLAGS AS PER TIME HISTORY AT THE OCCURRENCE OF A TIMING INTERRUPT (533 INTERRUPT)

卷之六

```

SUBROUTINE SET FLAGS
COMMON /SET6/  CLOCK(100), KIM(100), LOG(100)
COMMON /SET8/  CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOS(2),
-- I LN(2), IMAN(7,22), NSIM,NQ(2), ITQ(2,2), NPOP, ITPOP(2), XX(5,22),
-- XXX(5,22), YY(5,22), YYY(5,22), IMD(5)
COMMON /SET10/ NTOT, I TYPE
COMMON /SET12/ IDONE, I3, IM

```

SET CORRECT IMAGE FLAG TO MOVEMENT LOGIC VALUE AS PER HISTORY

```
10 IFLAG(KIM(I3))=LOG(I3)
```

```

I3=I3+1
ITOT=ITOT+1
IF (ITOT.GT.NTOT) IDONE=3: RETURN
IF (I3.GT.100) IDONE=2: RETURN

```

IF TIME TO NEXT EVENT IS SMALL, SET THE FLAG

```
IF(CLOCK(I3).LE.0.1) GO TO 10
```

SET MAX (HYBRID LIMITATION)

```
IF(CLOCK(I3).GT.99.)CLOCK(I3)=99.
```

SET THE TIMER FOR THE NEXT EVENT INTERRUPT

```
CALL DAC(1,CLOCK(I3)/100.)
CALL COMPUTE
RETURN
END
```



# SUBROUTINE FLAGS

PURPOSE: DISPLAY AND MOVE THE DYNAMIC IMAGES

```

SUBROUTINE FLAGS
COMMON /SET5/ IDEV, IER
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOS(2),
-ILN(2), IMAN(7,22), NSIM, NQ(2), ITQ(2,2), NPOP, ITPOP(2), XX(5,22),
-XXX(5,22), YY(5,22), YYY(5,22), IMD(5)
COMMON /SET10/ NTOI, ITYPE
COMMON /SET12/ IDONE, I3, IM
1 FORMAT(I4)

```

CHECK ALL THE FLAGS

```

IM=22
REPEAT 99 WHILE IM.GT.0
IF(IFLAG(IM).GT.0) GO TO(10,20,30,40,50,60,70,
-71,72,73,74,75) IFLAG(IM)
99 IM=IM-1
RETURN

```

HORIZONTAL MOVEMENT RIGHT (ARRIVALS, STRAIGHT DEPARTURES)

```

10 IF(XX(1,IM).NE.XXX(1,IM).OR.IM.LE.10) GO TO 5
IF(IM.GE.17) IM=IM-12; CALL REMAN; IM=IM+12; GO TO 5
IM=IM-6; CALL REMAN; IM=IM+6
5 CALL GRAPHO(IDEV,IMAN(1,IM),7,NSIM+IM,IER)
TEST=XX(1,IM)+CHNG(1)
IF(TFST.LE.ENDD(5,IM)) GO TO 18
11 IF(IM.GT.4) GO TO 13
12 IF(ITYPE.NE.3) GO TO 15
NPOP=NPOP-1; ENCODE(4,1,ITPOP) NPOP
CALL TEXTTO(IDEV,ITPOP,2,ILN(2),IPOS(2),3,3,IER)
NQ(1)=NQ(1)+1
IF(NQ(1).LE.0) GO TO 13
ENCODE(4,1,ITQ(1,1)) NQ(1)
CALL TEXTTO(IDEV,ITQ(1,1),2,ILN(1),IPOS(1),3,3,IER)
GO TO 13
15 IF(ITYPE.NE.5) IT=1; GO TO 6
IF(IM.EQ.1 .OR. IM.EQ.3) IT=1; GO TO 6
IT=2

```



```

6 NQ(IT)=NQ(IT)+1;
-IF(NQ(IT).GT.0) ENCODE(4,1,ITQ(1,IT)) NQ(IT);
-CALL TEXTQ (IDEV,ITQ(1,IT),2,ILN(IT),IPOS(IT),3,3,IER)
13 CALL REMAN: GO TO 99
18 DO 19 I=2,6
J=I-1
XX(J,IM)=XX(J,IM)+CHNG(1)
19 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
GO TO 99

C
C
C HORIZONTAL MOVEMENT RIGHT (WHEN A COURSE CHANGE WILL OCCUR)

20 IF(XX(1,IM).NE.XXX(1,IM).OR.IM.LE.10) GO TO 7
IF(IM.GE.17) IM=IM-12; CALL REMAN: IM=IM+12; GO TO 7
IM=IM-6; CALL REMAN: IM=IM+6
7 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)
TEST=XX(1,IM)+CHNG(3)
IF(TEST.GT.ENDD(1,IM)) IFLAG(IM)=3; GO TO 99
DO 28 I=2,6
J=I-1
XX(J,IM)=XX(J,IM)+CHNG(3)
28 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
GO TO 99

C
C
C VERTICAL MOVEMENT DOWNWARD

30 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)
TEST=YY(1,IM)-CHNG(3)
IF(TEST.GE.ENDD(2,IM)) GO TO 31
IF(IM.LE.4) CALL REMAN: GO TO 99
IFLAG(IM)=4; GO TO 99
DO 38 I=2,6
J=I-1
YY(J,IM)=YY(J,IM)-CHNG(3)
38 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
GO TO 99

C
C
C HORIZONTAL MOVEMENT TO THE LEFT

40 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)
TEST=XX(1,IM)-CHNG(3)*2.0
IF(TEST.GE.ENDD(3,IM)) GO TO 47
IFLAG(IM)=5; GO TO 99
DO 48 I=2,6
J=I-1
XX(J,IM)=XX(J,IM)-CHNG(3)*2.0
48 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
GO TO 99

```



```

C
C
C
      VERTICAL MOVEMENT UPWARD
50 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)
   TEST=YY(1,IM)+CHNG(4)
   IF(TEST.LE.ENDD(4,IM)) GO TO 57
   IF(ITYPE.NE.4) GO TO 55
   NQ(1)=NQ(1)+1
   IF(NQ(1).LE.0) GO TO 56
   FNCODE(4,1,ITQ(1,1)) NQ(1)
   CALL TEXTTO (IDEV,ITQ(1,1),2,ILN(1),IPOS(1),3,3,IER)
   GO TO 56
55 NPOP=NPOP+1; ENCODE(4,1,ITPOP) NPOP;
   -CALL TEXTTO (IDEV,ITPOP,2,ILN(2),IPOS(2),3,3,IER)
56 CALL REMAN: GO TO 99
57 DO 58 I=2,6
   J=I-1
   YY(J,IM)=YY(J,IM)+CHNG(4)
58 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
   GO TO 99
C
C
C
      MOVEMENT TO-SERVICE
60 IF(XX(1,IM).NE.XXX(1,IM)) GO TO 61
   IF(ITYPE.NE.5.OR. IM.LT.8) GO TO 65
   NQ(2)=NQ(2)-1
   IF(NQ(2).GE.0) ENCODE(4,1,ITQ(1,2)) NQ(2);
   -CALL TEXTTO (IDEV,ITQ(1,2),2,ILN(2),IPOS(2),3,3,IER)
   GO TO 61
65 NQ(1)=NQ(1)-1
   IF(NQ(1).LT.0) GO TO 61
   FNCODE(4,1,ITQ(1,1)) NQ(1)
   CALL TEXTTO (IDEV,ITQ(1,1),2,ILN(1),IPOS(1),3,3,IER)
61 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)
   TEST=XX(1,IM)+CHNG(2)
   IF(TEST.GT.ENDD(1,IM)) IFLAG(IM)=0; GO TO 99
   DO 68 I=2,6
   J=I-1
   XX(J,IM)=XX(J,IM)+CHNG(2)
   YY(J,IM)=YY(J,IM)+CHNG(IM)
68 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
   GO TO 99
C
C
C
      CAUSE DELAY
70 IFLAG(IM)=8 : GO TO 99
71 IFLAG(IM)=9 : GO TO 99
72 IFLAG(IM)=10: GO TO 99

```



```
73 IFLAG(IM)=11: GO TO 99
74 IFLAG(IM)=12: GO TO 99
75 IFLAG(IM)=6 : GO TO 99
    END
```



```

CCCCCCCC
SUBROUTINE REMAN
PURPOSE:  BLANK OUT IMAGES, AND REPOSITION
CCCCCCCC

```

```

SUBROUTINE REMAN
COMMON /SET5/ IDEV, IER
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOS(2),
-ILN(2), IMAN(7,22), NSIM,NQ(2), ITQ(2,2), NPOP, ITPOP(2), XX(5,22),
-XXX(5,22), YY(5,22), YYY(5,22), IMD(5)
COMMON /SET10/ NTOT, ITYPE
COMMON /SET12/ IDONE, I3, IM
IF (IM.LE.4 .OR. IM.GE.11) IFLAG(IM)=0
DO 10 K=2,6
10  IMAN(K,IM)=0
CALL GRAPHQ (IDEV, IMAN(1,IM),7, NSIM+IM, IER)
DO 20 K=2,6
J=K-1
XX(J,IM)=XXX(J,IM)
YY(J,IM)=YYY(J,IM)
20  IMAN(K,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
RETURN
END

```



```

CCCCCCCC
SUBROUTINE GGC
PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C MODEL

```

```

SUBROUTINE GGC
COMMON /SET1/ I SERV1(37), IPATH1(19), IPATH2(19), IPATH3(19),
- IPATH7(8)
COMMON /SET2/ I SERV2(19), IQUE1(7), IQUE2(12), IQUE3(7), IPATH4(10),
- IPATH5(10), IPATH9(7), IPATH10(5), IPATH11(4),
- IPATH12(4), IPATH13(6)
COMMON /SET3/ X(14), Y(25), MD(6), IC, IC1, NULL
COMMON /SET5/ IDEV, IER
DO 10 I=1, IC
10 MD(I)=1
CALL SERV1
CALL GRAPHO (IDEV, I SERV1, 6, IC+1, 1, IER)
CALL PATH1
CALL GRAPHO(IDEV, IPATH1, 3, IC+1, 2, IER)
CALL PATH2
CALL GRAPHO (IDEV, IPATH2, 3, IC+1, 3, IER)
CALL GRAPHO (IDEV, IPATH12, 4, 4, IER)
CALL GRAPHO (IDEV, IQUE1, 7, 5, IER)
RETURN
END

```







```

C
C
C
C
C
C
C
SUBROUTINE FEEDBACK
PURPOSE:  DISPLAY STATIC PICTURE FOR G/G/C WITH FEEDBACK

```

```

C
SUBROUTINE FEEDBK
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IQUE3(7),IPATH4(10),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/ IDEV, IER
CALL GGC
DO 10 I=1,4
10 MD(I)=1
IF(IC.LT.6)MD(1)=0;IF(IC.LT.4)MD(2)=0;IF(IC.LT.2)MD(3)=0
CALL PATH7
CALL GRAPHO (IDEV,IPATH7,8,6,IER)
CALL GRAPHO (IDEV,IPATH10,5,7,IER)
RETURN
END

```



```

CCCCCCCC
SUBROUTINE FINITE
PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C WITH FINITE SOURCE

```

```

SUBROUTINE FINITE
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IQUE3(7),IPATH4(10),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/ IDEV,IER
DO 10 I=1,4
10 MD(I)=1
IF(IC.LT.6)MD(1)=0;IF(IC.LT.4)MD(2)=0;IF(IC.LT.2)MD(3)=0
CALL PATH7
DO 11 I=1,IC
11 MD(I)=1
CALL SERV1
CALL PATH1
CALL PATH3
CALL GRAPH0 (IDEV, ISERV1,6, IC+1,1,IER)
CALL GRAPH0 (IDEV, IQUE1,7,2,IER)
CALL GRAPH0 (IDEV, IQUE3,7,3,IER)
CALL GRAPH0 (IDEV, IPATH1,3, IC+1,4,IER)
CALL GRAPH0 (IDEV, IPATH3,3, IC+1,5,IER)
CALL GRAPH0 (IDEV, IPATH7,8,6,IER)
CALL GRAPH0 (IDEV, IPATH9,7,7,IER)
RETURN
END

```







CCCCCCCC

SUBROUTINE SERV1

PURPOSE: PACK THE SERVICE BOXES

CCCCCCCC

```
SUBROUTINE SERV1
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
ISERV1(1)=IHEAD(0,10)
ISERV1(2)=IPACK(X(11),Y(13),0)
ISERV1(3)=IPACK(X(11),Y(17),MD(1))
ISERV1(4)=IPACK(X(12),Y(17),MD(1))
ISERV1(5)=IPACK(X(12),Y(13),MD(1))
ISERV1(6)=IPACK(X(11),Y(13),MD(1))
ISERV1(7)=0
ISERV1(8)=IPACK(X(11),Y(11),0)
ISERV1(9)=IPACK(X(11),Y(7),MD(2))
ISERV1(10)=IPACK(X(12),Y(7),MD(2))
ISERV1(11)=IPACK(X(12),Y(11),MD(2))
ISERV1(12)=IPACK(X(11),Y(11),MD(2))
ISERV1(13)=0
ISERV1(14)=IPACK(X(11),Y(18),0)
ISERV1(15)=IPACK(X(11),Y(20),MD(3))
ISERV1(16)=IPACK(X(12),Y(20),MD(3))
ISERV1(17)=IPACK(X(12),Y(18),MD(3))
ISERV1(18)=IPACK(X(11),Y(18),MD(3))
ISERV1(19)=0
ISERV1(20)=IPACK(X(11),Y(6),0)
ISERV1(21)=IPACK(X(11),Y(4),MD(4))
ISERV1(22)=IPACK(X(12),Y(4),MD(4))
ISERV1(23)=IPACK(X(12),Y(6),MD(4))
ISERV1(24)=IPACK(X(11),Y(6),MD(4))
ISERV1(25)=0
ISERV1(26)=IPACK(X(11),Y(21),0)
ISERV1(27)=IPACK(X(11),Y(23),MD(5))
ISERV1(28)=IPACK(X(12),Y(23),MD(5))
ISERV1(29)=IPACK(X(12),Y(21),MD(5))
ISERV1(30)=IPACK(X(11),Y(21),MD(5))
ISERV1(31)=0
ISERV1(32)=IPACK(X(11),Y(3),0)
ISERV1(33)=IPACK(X(11),Y(1),MD(6))
ISERV1(34)=IPACK(X(12),Y(1),MD(6))
```



```
ISERV1(35)=IPACK(X(12),Y(3),MD(6))  
ISERV1(36)=IPACK(X(11),Y(3),MD(6))  
ISERV1(37)=0  
RETURN  
END
```



CCCCCCCC

# SUBROUTINE PATH1

PURPOSE: PACK THE TO-SERVICE PATHS

```
SUBROUTINE PATH1
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
IPATH1(1)=IHEAD(1,8)
IPATH1(2)=IPACK(X(10),Y(12),0)
IPATH1(3)=IPACK(X(11),Y(15),MD(1))
IPATH1(4)=0
IPATH1(5)=IPACK(X(10),Y(12),0)
IPATH1(6)=IPACK(X(11),Y(9),MD(2))
IPATH1(7)=0
IPATH1(8)=IPACK(X(10),Y(12),0)
IPATH1(9)=IPACK(X(11),Y(19),MD(3))
IPATH1(10)=0
IPATH1(11)=IPACK(X(10),Y(12),0)
IPATH1(12)=IPACK(X(11),Y(5),MD(4))
IPATH1(13)=0
IPATH1(14)=IPACK(X(10),Y(12),0)
IPATH1(15)=IPACK(X(11),Y(22),MD(5))
IPATH1(16)=0
IPATH1(17)=IPACK(X(10),Y(12),0)
IPATH1(18)=IPACK(X(11),Y(2),MD(6))
IPATH1(19)=0
RETURN
END
```



CCCCCCCC

SUBROUTINE PATH2

PURPOSE: PACK THE HCRIZONTAL FROM-SERVICE PATHS

```

SUBROUTINE PATH2
COMMON /SET1/ISERV1(37), IPATH1(19), IPATH2(19), IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14), Y(25), MD(6), IC, IC1, NULL
IPATH2(1)=IHEAD(1,8)
IPATH2(2)=IPACK(X(12), Y(15), 0)
IPATH2(3)=IPACK(X(14), Y(15), MD(1))
IPATH2(4)=0
IPATH2(5)=IPACK(X(14), Y(9), 0)
IPATH2(6)=IPACK(X(12), Y(9), MD(2))
IPATH2(7)=0
IPATH2(8)=IPACK(X(12), Y(19), 0)
IPATH2(9)=IPACK(X(14), Y(19), MD(3))
IPATH2(10)=0
IPATH2(11)=IPACK(X(14), Y(5), 0)
IPATH2(12)=IPACK(X(12), Y(5), MD(4))
IPATH2(13)=0
IPATH2(14)=IPACK(X(12), Y(22), 0)
IPATH2(15)=IPACK(X(14), Y(22), MD(5))
IPATH2(16)=0
IPATH2(17)=IPACK(X(14), Y(2), 0)
IPATH2(18)=IPACK(X(12), Y(2), MD(6))
IPATH2(19)=0
RETURN
END

```



```

C
C
C
C
C
C
C
C
SUBROUTINE PATH3
PURPOSE:  PACK THE SHORT HORIZONTAL FROM-SERVICE PATHS

```

```

SUBROUTINE PATH3
COMMON /SET1/ISERV1(37), IPATH1(19), IPATH2(19), IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
IPATH3(1)=IHEAD(1,8)
IPATH3(2)=IPACK(X(12),Y(15),0)
IPATH3(3)=IPACK(X(13),Y(15),MD(1))
IPATH3(4)=0
IPATH3(5)=IPACK(X(13),Y(9),0)
IPATH3(6)=IPACK(X(12),Y(9),MD(2))
IPATH3(7)=0
IPATH3(8)=IPACK(X(12),Y(19),0)
IPATH3(9)=IPACK(X(13),Y(19),MD(3))
IPATH3(10)=0
IPATH3(11)=IPACK(X(13),Y(5),0)
IPATH3(12)=IPACK(X(12),Y(5),MD(4))
IPATH3(13)=0
IPATH3(14)=IPACK(X(12),Y(22),0)
IPATH3(15)=IPACK(X(13),Y(22),MD(5))
IPATH3(16)=0
IPATH3(17)=IPACK(X(13),Y(2),0)
IPATH3(18)=IPACK(X(12),Y(2),MD(6))
IPATH3(19)=0
RETURN
END

```



CCCCCCCC

SUBROUTINE PATH7

PURPOSE: PACK A VERTICAL PATH

```
SUBROUTINE PATH7
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14),Y(25),MD(6),IC,ICI, NULL
IPATH7(1)=IHEAD(1,8)
IPATH7(2)=IPACK(X(13),Y(2),0)
IPATH7(3)=IPACK(X(13),Y(5),MD(1))
IPATH7(4)=IPACK(X(13),Y(9),MD(2))
IPATH7(5)=IPACK(X(13),Y(15),MD(3))
IPATH7(6)=IPACK(X(13),Y(19),1)
IPATH7(7)=IPACK(X(13),Y(24),MD(4))
IPATH7(8)=0
RETURN
END
```



此詩蓋記於松林之遊也詩中言松林之遊者蓋記於松林之遊也

## PURPOSE: PACK OTHER BOXES AND PATHS

此後我輩無不為世所共知也

75



```

ISERV2(6)=IPACK(X(6),Y(18),1)
ISERV2(7)=0
ISERV2(8)=IPACK(X(6),Y(17),0)
ISERV2(9)=IPACK(X(6),Y(13),1)
ISERV2(10)=IPACK(X(4),Y(13),1)
ISERV2(11)=IPACK(X(4),Y(17),1)
ISERV2(12)=IPACK(X(6),Y(17),1)
ISERV2(13)=0
ISERV2(14)=IPACK(X(6),Y(21),0)
ISERV2(15)=IPACK(X(6),Y(23),1)
ISERV2(16)=IPACK(X(4),Y(23),1)
ISERV2(17)=IPACK(X(4),Y(21),1)
ISERV2(18)=IPACK(X(6),Y(21),1)
ISERV2(19)=0
IPATH4(1)=IHEAD(1,8)
IPATH4(2)=IPACK(X(10),Y(5),0)
IPATH4(3)=IPACK(X(11),Y(5),1)
IPATH4(4)=0
IPATH4(5)=IPACK(X(10),Y(5),0)
IPATH4(6)=IPACK(X(11),Y(9),1)
IPATH4(7)=0
IPATH4(8)=IPACK(X(10),Y(5),0)
IPATH4(9)=IPACK(X(11),Y(2),1)
IPATH4(10)=0
IPATH5(1)=IHEAD(1,8)
IPATH5(2)=IPACK(X(10),Y(19),0)
IPATH5(3)=IPACK(X(11),Y(19),1)
IPATH5(4)=0
IPATH5(5)=IPACK(X(10),Y(19),0)
IPATH5(6)=IPACK(X(11),Y(22),1)
IPATH5(7)=0
IPATH5(8)=IPACK(X(10),Y(19),0)
IPATH5(9)=IPACK(X(11),Y(15),1)
IPATH5(10)=0
IPATH9(1)=IHEAD(1,8)
IPATH9(2)=IPACK(X(13),Y(24),0)
IPATH9(3)=IPACK(X(5),Y(24),1)
IPATH9(4)=IPACK(X(5),Y(16),1)
IPATH9(5)=IPACK(X(7),Y(12),0)
IPATH9(6)=IPACK(X(9),Y(12),1)
IPATH9(7)=0
IPATH10(1)=IHEAD(1,8)
IPATH10(2)=IPACK(X(13),Y(24),0)
IPATH10(3)=IPACK(X(8),Y(24),1)
IPATH10(4)=IPACK(X(8),Y(12),1)
IPATH10(5)=0
IPATH11(1)=IHEAD(1,8)
IPATH11(2)=IPACK(X(8),Y(12),0)

```



```
IPATH11(3)=IPACK(X(8),Y(25),1)
IPATH11(4)=0
IPATH12(1)=IHEAD(1,8)
IPATH12(2)=IPACK(X(9),Y(12),0)
IPATH12(3)=IPACK(X(1),Y(12),1)
IPATH12(4)=0
IPATH13(1)=IHEAD(1,8)
IPATH13(2)=IPACK(X(9),Y(5),0)
IPATH13(3)=IPACK(X(1),Y(5),1)
IPATH13(4)=IPACK(X(1),Y(19),0)
IPATH13(5)=IPACK(X(9),Y(19),1)
IPATH13(6)=0
RETURN
END
```







YY(3,03)=YY(3,03)=Y(A3)+.05  
 XX(4,03)=XX(4,03)=X(A1)+.015  
 YY(4,03)=YY(4,03)=Y(A3)+.02  
 XX(5,03)=XX(5,03)=X(A1)-.02  
 YY(5,03)=YY(5,03)=Y(A3)-.05  
 XX(1,04)=XX(1,04)=X(A2)+.02  
 YY(1,04)=YY(1,04)=Y(A4)-.05  
 XX(2,04)=XX(2,04)=X(A2)-.015  
 YY(2,04)=YY(2,04)=Y(A4)+.02  
 XX(3,04)=XX(3,04)=X(A2)  
 YY(3,04)=YY(3,04)=Y(A4)+.05  
 XX(4,04)=XX(4,04)=X(A2)+.015  
 YY(4,04)=YY(4,04)=Y(A4)+.02  
 XX(5,04)=XX(5,04)=X(A2)-.02  
 YY(5,04)=YY(5,04)=Y(A4)-.05  
 XX(1,05)=XX(1,05)=X(G1)+.02  
 YY(1,05)=YY(1,05)=Y(G3)-.05  
 XX(2,05)=XX(2,05)=X(G1)-.015  
 YY(2,05)=YY(2,05)=Y(G3)+.02  
 XX(3,05)=XX(3,05)=X(G1)  
 YY(3,05)=YY(3,05)=Y(G3)+.05  
 XX(4,05)=XX(4,05)=X(G1)+.015  
 YY(4,05)=YY(4,05)=Y(G3)+.02  
 XX(5,05)=XX(5,05)=X(G1)-.02  
 YY(5,05)=YY(5,05)=Y(G3)-.05  
 XX(1,06)=XX(1,06)=X(G1)+.02  
 YY(1,06)=YY(1,06)=Y(G3)-.05  
 XX(2,06)=XX(2,06)=X(G1)-.015  
 YY(2,06)=YY(2,06)=Y(G3)+.02  
 XX(3,06)=XX(3,06)=X(G1)  
 YY(3,06)=YY(3,06)=Y(G3)+.05  
 XX(4,06)=XX(4,06)=X(G1)+.015  
 YY(4,06)=YY(4,06)=Y(G3)+.02  
 XX(5,06)=XX(5,06)=X(G1)-.02  
 YY(5,06)=YY(5,06)=Y(G3)-.05  
 XX(1,07)=XX(1,07)=X(G1)+.02  
 YY(1,07)=YY(1,07)=Y(G3)-.05  
 XX(2,07)=XX(2,07)=X(G1)-.015  
 YY(2,07)=YY(2,07)=Y(G3)+.02  
 XX(3,07)=XX(3,07)=X(G1)  
 YY(3,07)=YY(3,07)=Y(G3)+.05  
 XX(4,07)=XX(4,07)=X(G1)+.015  
 YY(4,07)=YY(4,07)=Y(G3)+.02  
 XX(5,07)=XX(5,07)=X(G1)-.02  
 YY(5,07)=YY(5,07)=Y(G3)-.05  
 XX(1,08)=XX(1,08)=X(G2)+.02  
 YY(1,08)=YY(1,08)=Y(G4)-.05  
 XX(2,08)=XX(2,08)=X(G2)-.015



YY(2,08)=YY(2,08)=Y(G4)+.02  
 XX(3,08)=XX(3,08)=X(G2)  
 YY(3,08)=YY(3,08)=Y(G4)+.05  
 XX(4,08)=XX(4,08)=X(G2)+.015  
 YY(4,08)=YY(4,08)=Y(G4)+.02  
 XX(5,08)=XX(5,08)=X(G2)+.02  
 YY(5,08)=YY(5,08)=Y(G4)+.05  
 XX(1,09)=XX(1,09)=X(G2)+.02  
 YY(1,09)=YY(1,09)=Y(G4)+.05  
 XX(2,09)=XX(2,09)=X(G2)+.015  
 YY(2,09)=YY(2,09)=Y(G4)+.02  
 XX(3,09)=XX(3,09)=X(G2)+.05  
 YY(3,09)=YY(3,09)=Y(G4)+.015  
 XX(4,09)=XX(4,09)=X(G2)+.02  
 YY(4,09)=YY(4,09)=Y(G4)+.05  
 XX(5,09)=XX(5,09)=X(G2)+.02  
 YY(5,09)=YY(5,09)=Y(G4)+.05  
 XX(1,10)=XX(1,10)=X(G2)+.02  
 YY(1,10)=YY(1,10)=Y(G4)+.05  
 XX(2,10)=XX(2,10)=X(G2)+.015  
 YY(2,10)=YY(2,10)=Y(G4)+.02  
 XX(3,10)=XX(3,10)=X(G2)+.05  
 YY(3,10)=YY(3,10)=Y(G4)+.015  
 XX(4,10)=XX(4,10)=X(G2)+.02  
 YY(4,10)=YY(4,10)=Y(G4)+.05  
 XX(5,10)=XX(5,10)=X(G2)+.02  
 YY(5,10)=YY(5,10)=Y(G4)+.05  
 XX(1,11)=XX(1,11)=X(S1)+.02  
 YY(1,11)=YY(1,11)=Y(S3)+.05  
 XX(2,11)=XX(2,11)=X(S1)+.015  
 YY(2,11)=YY(2,11)=Y(S3)+.02  
 XX(3,11)=XX(3,11)=X(S1)+.05  
 YY(3,11)=YY(3,11)=Y(S3)+.015  
 XX(4,11)=XX(4,11)=X(S1)+.02  
 YY(4,11)=YY(4,11)=Y(S3)+.05  
 XX(5,11)=XX(5,11)=X(S1)+.02  
 YY(5,11)=YY(5,11)=Y(S3)+.05  
 XX(1,12)=XX(1,12)=X(S1)+.02  
 YY(1,12)=YY(1,12)=Y(S4)+.05  
 XX(2,12)=XX(2,12)=X(S1)+.015  
 YY(2,12)=YY(2,12)=Y(S4)+.02  
 XX(3,12)=XX(3,12)=X(S1)+.05  
 YY(3,12)=YY(3,12)=Y(S4)+.015  
 XX(4,12)=XX(4,12)=X(S1)+.02  
 YY(4,12)=YY(4,12)=Y(S4)+.05  
 XX(5,12)=XX(5,12)=X(S1)+.02  
 YY(5,12)=YY(5,12)=Y(S4)+.05  
 XX(1,13)=XX(1,13)=X(S1)+.02



YY(1,13)=YY(1,13)=Y(S5)-.05  
 XX(2,13)=XX(2,13)=X(S1)-.015  
 YY(2,13)=YY(2,13)=Y(S5)+.02  
 XX(3,13)=XX(3,13)=X(S1)  
 YY(3,13)=YY(3,13)=Y(S5)+.05  
 XX(4,13)=XX(4,13)=X(S1)+.015  
 YY(4,13)=YY(4,13)=Y(S5)+.02  
 XX(5,13)=XX(5,13)=X(S1)-.02  
 YY(5,13)=YY(5,13)=Y(S5)-.05  
 XX(1,14)=XX(1,14)=X(S2)+.02  
 YY(1,14)=YY(1,14)=Y(S6)-.05  
 XX(2,14)=XX(2,14)=X(S2)-.015  
 YY(2,14)=YY(2,14)=Y(S6)+.02  
 XX(3,14)=XX(3,14)=X(S2)  
 YY(3,14)=YY(3,14)=Y(S6)+.05  
 XX(4,14)=XX(4,14)=X(S2)+.015  
 YY(4,14)=YY(4,14)=Y(S6)+.02  
 XX(5,14)=XX(5,14)=X(S2)-.02  
 YY(5,14)=YY(5,14)=Y(S6)-.05  
 XX(1,15)=XX(1,15)=X(S2)+.02  
 YY(1,15)=YY(1,15)=Y(S7)-.05  
 XX(2,15)=XX(2,15)=X(S2)-.015  
 YY(2,15)=YY(2,15)=Y(S7)+.02  
 XX(3,15)=XX(3,15)=X(S2)  
 YY(3,15)=YY(3,15)=Y(S7)+.05  
 XX(4,15)=XX(4,15)=X(S2)+.015  
 YY(4,15)=YY(4,15)=Y(S7)+.02  
 XX(5,15)=XX(5,15)=X(S2)-.02  
 YY(5,15)=YY(5,15)=Y(S7)-.05  
 XX(1,16)=XX(1,16)=X(S2)+.02  
 YY(1,16)=YY(1,16)=Y(S8)-.05  
 XX(2,16)=XX(2,16)=X(S2)-.015  
 YY(2,16)=YY(2,16)=Y(S8)+.02  
 XX(3,16)=XX(3,16)=X(S2)  
 YY(3,16)=YY(3,16)=Y(S8)+.05  
 XX(4,16)=XX(4,16)=X(S2)+.015  
 YY(4,16)=YY(4,16)=Y(S8)+.02  
 XX(5,16)=XX(5,16)=X(S2)-.02  
 YY(5,16)=YY(5,16)=Y(S8)-.05  
 XX(1,17)=XX(1,17)=X(S1)+.02  
 YY(1,17)=YY(1,17)=Y(S3)-.05  
 XX(2,17)=XX(2,17)=X(S1)-.015  
 YY(2,17)=YY(2,17)=Y(S3)+.02  
 XX(3,17)=XX(3,17)=X(S1)  
 YY(3,17)=YY(3,17)=Y(S3)+.05  
 XX(4,17)=XX(4,17)=X(S1)+.015  
 YY(4,17)=YY(4,17)=Y(S3)+.02  
 XX(5,17)=XX(5,17)=X(S1)-.02



YYY(5,17)=YY(5,17)=Y(S3)-.05  
 XXX(1,18)=XX(1,18)=X(S1)+.02  
 YYY(1,18)=YY(1,18)=Y(S4)-.05  
 XXX(2,18)=XX(2,18)=X(S1)-.015  
 YYY(2,18)=YY(2,18)=Y(S4)+.02  
 XXX(3,18)=XX(3,18)=X(S1)+.05  
 YYY(3,18)=YY(3,18)=Y(S4)+.015  
 XXX(4,18)=XX(4,18)=X(S1)+.02  
 YYY(4,18)=YY(4,18)=Y(S4)-.02  
 XXX(5,18)=XX(5,18)=X(S1)-.05  
 YYY(5,18)=YY(5,18)=Y(S4)+.02  
 XXX(1,19)=XX(1,19)=X(S1)+.05  
 YYY(1,19)=YY(1,19)=Y(S5)-.015  
 XXX(2,19)=XX(2,19)=X(S1)-.02  
 YYY(2,19)=YY(2,19)=Y(S5)+.02  
 XXX(3,19)=XX(3,19)=X(S1)+.05  
 YYY(3,19)=YY(3,19)=Y(S5)+.015  
 XXX(4,19)=XX(4,19)=X(S1)+.02  
 YYY(4,19)=YY(4,19)=Y(S5)-.02  
 XXX(5,19)=XX(5,19)=X(S1)-.05  
 YYY(5,19)=YY(5,19)=Y(S5)+.02  
 XXX(1,20)=XX(1,20)=X(S2)+.05  
 YYY(1,20)=YY(1,20)=Y(S6)-.015  
 XXX(2,20)=XX(2,20)=X(S2)+.02  
 YYY(2,20)=YY(2,20)=Y(S6)+.05  
 XXX(3,20)=XX(3,20)=X(S2)+.015  
 YYY(3,20)=YY(3,20)=Y(S6)+.02  
 XXX(4,20)=XX(4,20)=X(S2)-.05  
 YYY(4,20)=YY(4,20)=Y(S6)+.02  
 XXX(5,20)=XX(5,20)=X(S2)-.02  
 YYY(5,20)=YY(5,20)=Y(S6)-.05  
 XXX(1,21)=XX(1,21)=X(S2)+.02  
 YYY(1,21)=YY(1,21)=X(S7)-.015  
 XXX(2,21)=XX(2,21)=X(S2)-.02  
 YYY(2,21)=YY(2,21)=Y(S7)+.02  
 XXX(3,21)=XX(3,21)=X(S2)+.05  
 YYY(3,21)=YY(3,21)=Y(S7)+.015  
 XXX(4,21)=XX(4,21)=X(S2)+.02  
 YYY(4,21)=YY(4,21)=Y(S7)-.02  
 XXX(5,21)=XX(5,21)=X(S2)-.05  
 YYY(5,21)=YY(5,21)=Y(S7)+.02  
 XXX(1,22)=XX(1,22)=X(S2)+.05  
 YYY(1,22)=YY(1,22)=Y(S8)-.015  
 XXX(2,22)=XX(2,22)=X(S2)+.02  
 YYY(2,22)=YY(2,22)=Y(S8)+.05  
 XXX(3,22)=XX(3,22)=X(S2)+.015  
 YYY(3,22)=YY(3,22)=Y(S8)+.02  
 XXX(4,22)=XX(4,22)=X(S2)+.05



```

YYY(4,22)=YY(4,22)=Y(S8)+.02
XXX(5,22)=XX(5,22)=X(S2)-.02
YYY(5,22)=YY(5,22)=Y(S8)-.05
DO 20 I=1,22
DO 20 J=2,6
K=J-1
20 IMAN(J,I)=IPACK(XXX(K,I),YYY(K,I),IMD(K))
RETURN
END

```



```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE GINP
WRITTEN BY A. WONG, EE DEPT., USNPGS
PURPOSE: PROVIDE NAMELIST INPUT FROM GRAPHICS TERMINAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

1      SUBROUTINE GINP(IDEV,ITDIR,IBLK,IBUF)
2      DIMENSION IBUF(1),ITDIR(1)
3      IB=IBLK+1
4      NULL=-1
5      IF(IBUF(1))1,50,100
6      IF(IBUF(1).NE.-1)GO TO 100
7      ENCODE(16,10,IBUF)
8      FORMAT('NAMELIST INPUT')
9      CALL TEXTO(IDEV,IBUF,4,18,28,3,3,IER)
10     IF(IER.NE.0)OUTPUT(101)IER,'GINP1'
11     RETURN
12     CALL TEXTO(IDEV,NULL,1,18,28,3,3,IER)
13     IF(IER.NE.0)OUTPUT(101)IER,'NULL1'
14     CALL TEXTO(IDEV,NULL,1,24,43,3,3,IER)
15     IF(IER.NE.0)OUTPUT(101)IER,'NULL2'
16     RETURN
17     CALL TEXTR(IDEV,NULL,1,24,43,3,3,IER)
18     IF(IER.NE.0)OUTPUT(101)IER,'GINP2'
19     IF(MOD(ITDIR(IB),8).EQ.0)GO TO 110
20     CALL TEXTI(IDEV,IBUF,24,0,IB,IER)
21     IF(IER.NE.0)OUTPUT(101)IER,'GINP3'
22     RETURN
23     END

```



SUBROUTINE GINPUT  
WRITTEN BY A. WONG, EE DEPT., USNPGS  
PURPOSE: PROVIDE NAMELIST INPUT FROM GRAPHICS TERMINAL

```

0 $GINPUT PZE  

BRM  

PZE  

PZE  

PZE  

LDP  

STD  

LDA  

STA  

LDA  

ADD  

STA  

LDA  

STA  

BRM  

PZE  

LDA  

ADD  

STA  

LDA  

XMA  

STA  

LDA  

BRM  

STZ  

BRM  

PZE  

LDA  

XMA  

BRR  

IDEV  

ITDIR  

IBLK  

INPADR  

0 9SETUPN  

3 0  

0 0  

0 IDEV  

TEXT033  

*IBLK  

BLCK INPADR  

BUF  

IBUF R>CNES  

*IBUF  

TEXT0  

0 INPADR  

READ  

PATCH  

BRM  

*PATCH  

BRM  

#101  

*IBUF  

9INPUT  

*IBUF  

TEXT0  


0 BRM  

*PATCH  

BRM  

GINPUT

```

	
TEXTO	PZE BRM
	O GINP
DKN	GINP



PZE	4	
PZE	0	
PZE	0	BLOCK
PZE	0	TEXT0
MPO		TEXT0
BRR		

IBUF		
	☆	
	☆	
	☆	
BLOCK	0	
PATCH	0	TEXT0
BRM		0563
READ		0773
BUF		
		\$



# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Chief of Naval Personnel Pers 11b Department of the Navy Washington, D. C. 20370	1
4. Naval Postgraduate School Department of Operations Research and Administrative Sciences Monterey, California 93940	1
5. Professor R. W. Butterworth, Code 55 Bd Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6. Professor W. M. Raike, Code 55 Rj Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7. Professor P. A. W. Lewis, Code 55 Lw Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8. Professor D. P. Gaver, Code 55 Gv Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
9. Professor A. F. Andrus, Code 55 As Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1



10. Lieutenant Cyrus Michael Riddell  
1615 92 North East  
Bellevue, Washington 98004

1



UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE A Queuing Model Simulation with Dynamic Graphical Display			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Master's Thesis; March 1973			
5. AUTHOR(S) (First name, middle initial, last name) Cyrus Michael Riddell			
6. REPORT DATE March 1973		7a. TOTAL NO. OF PAGES 90	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	

13. ABSTRACT <p>The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.</p>
--



14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
<p>Queues</p> <p>Simulation</p> <p>Computer Graphics</p>						



2 JUN 76

24425

Thesis

143254

R425 Riddell

c.1.

A queuing model simulation with dynamic graphical display.

2 JUN 76

24425

Thesis

143254

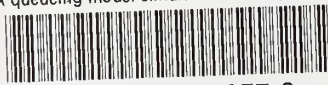
R425 Riddell

c.1.

A queuing model simulation with dynamic graphical display.

thesR425

A queueing model simulation with dynamic



3 2768 000 99057 6

DUDLEY KNOX LIBRARY